TRƯỜNG ĐẠI HỌC HỒNG ĐỨC

KHOA CNTT & TT

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN NĂM HỌC 2022 – 2023

NGHIÊN CỨU ỨNG DỤNG CÁC KĨ THUẬT PHÂN LỚP DỮ LIỆU NHẰM PHÁT HIỆN HÌNH THỨC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN (DDoS)

THANH HÓA, THÁNG 04/2023

TRƯỜNG ĐẠI HỌC HỒNG ĐỨC

KHOA CNTT & TT

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN NĂM HỌC 2022 – 2023

NGHIÊN CỨU ỨNG DỤNG CÁC KĨ THUẬT PHÂN LỚP DỮ LIỆU NHẰM PHÁT HIỆN HÌNH THỨC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN (DDoS)

Sinh viên thực hiện:Lê Xuân Quang

Lê Đình Thắng

Hoàng Văn Huy

Cao Sơn Đăng

Giảng viên hướng dẫn: Nguyễn Thế Cường

THANH HÓA, THÁNG 04/2023

LỜI CẢM ƠN

Trước tiên, chúng tôi muốn gửi lời cảm ơn chân thành tới các giảng viên của Khoa Công nghệ thông tin và Truyền thông tại trường Đại học Hồng Đức. Nhờ sự truyền đạt của các thầy cô, chúng tôi đã tiếp thu được những kiến thức nền tảng và những kinh nghiệm thực tế hữu ích trong quá trình học tập và nghiên cứu.

Đặc biệt, chúng tôi muốn gửi lời cảm ơn đến TS Nguyễn Thế Cường, người đã hướng dẫn và giúp đỡ chúng tôi trong quá trình thực hiện đề tài. Nhờ những định hướng, giải đáp thắc mắc và lời khuyên của ông, chúng tôi đã có thể hoàn thành đề tài một cách tốt nhất.

Dù đã cố gắng hết sức, chúng tôi vẫn còn nhiều điều cần phải học hỏi và hoàn thiện. Trong quá trình thực hiện đề tài, chúng tôi không tránh khỏi những sai sót do kiến thức và kinh nghiệm còn hạn chế. Do đó, chúng tôi rất mong nhận được sự góp ý, chỉ bảo từ các giảng viên để có thể hoàn thiện đề tài và cải thiện khả năng nghiên cứu của mình.

Một lần nữa, chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy cô!

Thanh Hóa, ngày tháng 4 năm 2023

Nhóm thực hiện

Lê Xuân Quang

LỜI NÓI ĐẦU

Trong lịch sử văn minh của loại người, đã trải qua nhiều thời kỳ như: thời đồ đá, thời phong kiến… Mỗi thời kỳ, con người chúng ta càng phát triển mạnh mẽ, mỗi thời kỳ đều có thành tựu nổi bật. Và đến ngày hôm nay, trải qua hơn 4000 năm văn minh nhân loại, chúng ta đang thực hiện cuộc cách mạng công nghiệp lần thứ 4. Các cuộc cách mạng công trước đã mang lại thành tựu vượt bậc cho con người trong nông nghiệp, công nghiệp… Khi nhắc đến cuộc cách mạng lần thứ 4, các chủ đề được đề cao phát triển mạnh mẽ đó là: dữ liệu lớn, trí tuệ nhân tạo, công nghệ Blockchain… Các lĩnh vực này đề thuộc ngành công nghệ thông tin, khi công nghệ ngày càng phát triển thì nguy cơ tấn công trên mạng Internet ngày càng cao và nó là thực trạng nhức nhối từ khi khai sinh ra Internet.

Trên phạm vi toàn cầu, năm 2022 tội phạm mạng gây thiệt hại lên tới hơn 1.000 tỷ USD mỗi năm, tương đương 1,18% GDP toàn cầu. Còn ở Việt Nam, con số này là 883 triệu USD (tương đương 0,24% GDP). Từ những số liệu trên cho thấy tổn thất do tấn công mạng ảnh hưởng rất lớn đến nền kinh tế Việt Nam và trên toàn cầu.

Trong các loại tấn công mạng thì nổi bật lên đó là tấn công từ chối dịch vụ phân tán(DDoS). Vào năm 2022, số vụ tấn công DDoS đã tăng 150% trên toàn cầu so với năm trước. Trong khi tổng khối lượng tấn công toàn cầu được ghi nhận vào năm 2022 là 4,44PB, tăng 32% so với năm 2021. Nhận thấy các tác hại sâu sắc cảu tấn công từ chối dịch vụ phân tán(DDoS) nên trong đề tài này, chúng tôi chọn đề tài nghiên cứu: "**Nghiên cứu ứng dụng các kỹ thuật phân lớp dữ liệu nhằm phát hiện hình thức tấn công từ chối dịch vụ phân tán (DDoS)**". Ứng dụng đề tài giúp phân lớp các dữ liệu trong bộ dữ liệu tấn công DDoS dựa vào các phương pháp phân lớp dữ liệu một cách nhanh, chính xác hơn.

MỤC LỤC

MỞ ĐẦU **8**
1.1. Tính cấp thiết của đề tài 8
1.2 Mục tiêu đề tài 8
1.3. Đối tượng và phạm vi nghiên cứu 8
1.3.1. Đối tượng nghiên cứu 8
1.3.2. Phạm vi nghiên cứu 8
1.4. Phương pháp nghiên cứu 8
1.5. Nội dung nghiên cứu 8
CHƯƠNG 1: TỔNG QUAN VỀ TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN .9
1.1. Tổng quan về tấn công từ chối dịch vụ phân tán(DDoS) 9
1.2. Nguyên nhân DDOS có thể thực hiện được 10
1.2. Các giai đoạn của một cuộc tấn công DDoS 10
1.3. Phân loại tấn công từ chối dịch vụ phân tán(DDoS) 11
1.4. Những hình thức tấn công DDoS thường gặp phải 12
1.4.1. SYN Flood 12
1.4.2. UDP Flood 14
1.4.3. HTTP Flood 16
1.4.4. Ping of Death 17
1.4.5. Smurf Attack 18
1.4.6. Fraggle Attack 19
1.4.7. Slowloris 20
1.4.8. NTP Amplification 21
1.4.9. Advanced persistent Dos (APDos) 21
CHƯƠNG 2: TỔNG QUAN VỀ PHÂN LỚP DỮ LIỆU 22
2.1. Giới thiệu về phân lớp dữ liệu 22
2.2. Các vấn đề liên quan đến phân lớp dữ liệu 24
2.2.1. Chuẩn bị dữ liệu cho việc phân lớp 24
2.2.2. So sánh các mô hình phân lớp 24
2.3. Các phương thuật toán phân lớp dữ liệu 25
2.3.1. Thuật toán SVM 25
2.3.2. Thuật toán cây quyết định(Decision Tree) 26

2.3.3. Logistic Regression (Hồi quy Logistic) 32
1.2.4. Thuật toán rừng cây ngẫu nhiên(Ramdom forest) 36
1.2.5. Thuật toán K-Nearest Neighbors.....38
CHƯƠNG 3: ỨNG DỤNG CÁC KÝ THUẬT PHÂN LỚP DỮ LIỆU NHẰM PHÁT HIỆN HÌNH THỨC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN(DDoS) 42
3.1. Giới thiệu.....42
3.2. Xây dựng bộ dữ liệu.....42
3.2.1. Mô tả bộ dữ liệu 42
3.2.2. Phân tích đặc trưng trong cơ sở dữ liệu 45
3.2.3. Trích chọn đặc trưng 50
3.3. Triển khai các mô hình không sử dụng giải thuật lựa chọn đặc trưng 54
3.3. Triển khai các mô hình có sử dụng giải thuật lựa chọn đặc trưng.....55
KẾT LUẬN.....57
1. Kết quả đạt được.....57
2. Hạn chế.....57
3. Hướng phát triển.....57

DANH MỤC HÌNH ẢNH

Hình 1 : Mô hình tấn công DDoS 9
Hình 2 : Tấn công UDP Flood 14
Hình 3: Quá trình tấn công UDP Flood Attack 15
Hình 4: Tấn công HTTP Flood 16
Hình 5: Mô tả tấn công Ping of Death 17
Hình 6: Mô tả Smurf Attack.....18
Hình 7: Mô tả tấn công Slowloris 20
Hình 8: Bước xây dựng mô hình phân lớp 22
Hình 9: (b1)Ước lượng độ chính xác của mô hình.....23
Hình 10: (b2)Phân lớp dữ liệu mới 24
Hình 11: Ví dụ về cây quyết định 26
Hình 12: Hàm Entropy 29
Hình 13: Đồ thị phương trình hồi quy logistic.....34
Hình 14: Sơ đồ hoạt động của thuật toán Random Forest.....36
Hình 15: Sơ đồ thuật toán KNN.....39
Hình 16: Tỉ lệ phần trăm của những yêu cầu Bình thường và Độc hại trong cơ sở dữ liệu 44
Hình 17: Các bản ghi trong cơ sở dữ liệu 45
Hình 18: Biểu đồ các đặc trưng rỗng 45
Hình 19: Biểu đồ phân phối của các địa chỉ gửi và số lượng yêu cầu được gửi đi.....46
Hình 20: Biểu đồ về số lượng các yêu cầu có mục đích tấn công đối với các địa chỉ gửi 46
Hình 21: Biểu đồ về số lường các yêu cầu gửi từ các địa chỉ khác nhau.....46
Hình 22 : Biểu đồ phân bố về các yêu cầu và các giao thức được sử dụng 47
Hình 23: Biểu đồ phân bố về thời gian của các yêu cầu.....47
Hình 24: Biểu đồ phân bố số lượng các yêu cầu và kích thước dữ liệu truyền đi 48
Hình 25: Biểu đồ phân bố số lượng các yêu cầu và băng thông dữ liệu 48
Hình 26: Biểu đồ phân bố số lượng các yêu cầu và bộ chuyển mạch.....49
Hình 27: Biểu đồ phân bố số lượng các yêu cầu và số lượng gói tin.....49
Hình 28: Biểu đồ phân bố số lượng các yêu cầu và số byte dữ liệu.....50
Hình 29: Trọng số các đặc trưng được tính theo giải thuật NCA 52
Hình 30: Mức độ tương quan giữa các đặc trưng đã được lựa chọn trọng cơ sở dữ liệu 53

DANH MỤC BẢNG BIỂU

Bảng 1: So sánh cây quyết định và rừng cây ngẫu nhiên.....38
Bảng 2: Mô tả bộ dữ liệu.....44
Bảng 3: Trọng số được tính theo NCA.....52
Bảng 4:Trọng số các đặc trưng còn lại sau khi loại bỏ 53

DANH MỤC TỪ VIẾT TẮT

STT	Chữ viết tắt	Chữ viết đầy đủ
1	ICMP	Giao thức Thông báo Kiểm soát Internet
2	UDP	Giao thức gói dữ liệu người dùng
3	SYN	Tấn công nửa vời
4	TCP	Giao thức điều khiển truyền dẫn
5	DNS	Hệ Thống Tên Miền
6	SNMP	Giao thức giám sát mạng đơn giản
7	PDOS	Tên miền lừa đảo qua HTTPS
8	Botnet	Mạng bot
9	IPX	Trao đổi giao thức internet
10	CART	Cây phân loại và hồi quy
11	SVM	Máy véc tơ hỗ trợ
12	SOP	Quy trình vận hành tiêu chuẩn
13	KNN	K – láng giềng
14	SDN	Mạng được xác định bằng phần mềm
15	NCA	Phân tích thành phần lân cận

MỞ ĐẦU

1.1. Tính cấp thiết của đề tài

DDos là một hình thức tấn công nguy hiểm và gây tổn thất nặng nề cho các doanh nghiệp, tổ chức và cả người dùng cá nhân trên khắp thế giới.

Các cuộc tấn công DDos có thể gây ra hậu quả nghiêm trọng, từ làm gián đoạn dịch vụ trực tuyến, gây thiệt hại tài chính, đến việc đe dọa tính mạng và an ninh thông tin của người dùng. Các cuộc tấn công DDos cũng có thể được sử dụng như một công cụ trong các cuộc tấn công mạng khác, như trộm cắp dữ liệu hoặc phá hoại hệ thống.

Chính vì thế, chúng tôi chọn đề tài: "**Nghiên cứu ứng dụng các kỹ thuật phân lớp dữ liệu nhằm phát hiện hình thức tấn công từ chối dịch vụ phân tán (DDoS)**", sử dụng công nghệ thông minh để phát hiện các mẫu tấn công mới,hay triển khai các cơ chế đối phó nhanh chóng khi bị tấn công.

1.2 Mục tiêu đề tài

- Xây dựng thành công mô hình sử dụng các kỹ thuật phân lớp dữ liệu nhằm phát hiện tấn công DDoS.
- Ứng dụng trong thực tế.

1.3. Đối tượng và phạm vi nghiên cứu

1.3.1. Đối tượng nghiên cứu

Ngôn ngữ lập trình Python, các thuật toán phân lớp dữ liệu và tấn công từ chối dịch vụ phân tán(DDoS).

1.3.2. Phạm vi nghiên cứu

Dữ liệu về các tấn công dạng DDoS trong mạng các dịch vụ.

1.4. Phương pháp nghiên cứu

- Phương pháp chuyên gia.
- Phương pháp phân tích và tổng hợp lý thuyết.
- Phương pháp thực nghiệm.

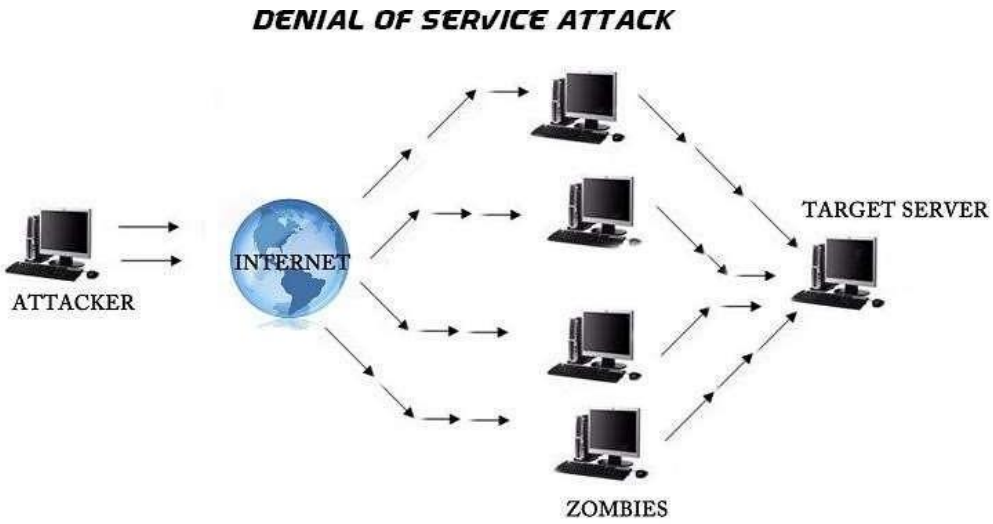
1.5. Nội dung nghiên cứu

- Nghiên cứu cơ sở lý thuyết về hình thức tấn công từ chối dịch vụ DDoS.
- Nghiên cứu về các kỹ thuật phân lớp dữ liệu.
- Nghiên cứu cơ sở dữ liệu mẫu về hình thức tấn công DDoS đã có.
- Xây dựng mô hình phân lớp dữ liệu dùng đề phát hiện hình thức tấn công DDoS dựa vào dữ liệu mẫu đã có.
- Đánh giá hiệu quả của mô hình.

CHƯƠNG 1: TỔNG QUAN VỀ TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN

1.1. Tổng quan về tấn công từ chối dịch vụ phân tán(DDoS)

Tấn công từ chối dịch vụ (DDoS - Distributed Denial of Service) là một loại tấn công mạng, mà trong đó một số lượng lớn các yêu cầu được gửi đến một máy chủ hay một hệ thống mạng, nhằm làm quá tải hệ thống đó, khiến cho nó không còn khả năng hoạt động bình thường hoặc không thể truy cập được từ bên ngoài.



Hình 1 : Mô hình tấn công DDoS

Tấn công DDoS tận dụng giới hạn dung lượng đặc biệt của các tài nguyên mạng, chẳng hạn như cơ sở hạ tầng hỗ trợ trang Web của công ty. Kẻ tấn công sẽ tạo ra nhiều yêu cầu đến tài nguyên Web bị tấn công, vượt quá khả năng xử lý và gây ra sự cố, làm ngắt kết nối trang web, khiến cho trang Web không thể hoạt động bình thường.

DDoS attack sử dụng các botnet (một mạng các máy tính bị lây nhiễm bởi phần mềm độc hại) để thực hiện tấn công.

Bằng cách sử dụng các botnet này, kẻ tấn công có thể kiểm soát một lượng lớn máy tính từ xa. Và sử dụng chúng để gửi lưu lượng truy cập giả mạo đến hệ thống đích. Khi lưu lượng truy cập tăng đột ngột và vượt quá khả năng xử lý của hệ thống đích. Hệ thống đó sẽ bị quá tải và không thể phục vụ các yêu cầu từ người dùng.

Năm 1999, loại tấn công DDoS xuất hiện lần đầu tiên và có sức mạnh cao hơn rất nhiều so với tấn công DoS truyền thống, bởi vì nguồn tấn công đến từ nhiều máy tính chứ không phải chỉ một máy tính như tấn công DoS. Hầu hết các cuộc tấn công DDoS được thực hiện để chiếm dụng băng thông, gây nghẽn mạng và đẩy hệ thống vào trạng thái ngừng hoạt động. Tuy nhiên, với sự phát triển của các thiết bị phần cứng và các hệ

thống phòng thủ, các hình thức tấn công DDoS ngày càng trở nên phức tạp và tinh vi, không chỉ giới hạn ở việc chiếm dụng băng thông mà còn khai thác các lỗ hổng trong các ứng dụng để tấn công, gây cạn kiệt tài nguyên của hệ thống. Các loại tấn công này được đánh giá là nguy hiểm hơn, vì chúng có thể gây tổn hại trực tiếp đến cơ sở dữ liệu.

1.2. Nguyên nhân DDOS có thể thực hiện được

Thiết kế Internet hiện tại tập trung vào tính hiệu quả trong việc di chuyển các gói tin từ nguồn đến đích. Thiết kế này tuân theo mô hình end-to-end, nghĩa là mạng trung gian cung cấp dịch vụ chuyển tiếp gói tin tối thiểu, tốt nhất, để người gửi và người nhận triển khai các giao thức tiên tiến để đạt được các đảm bảo dịch vụ mong muốn như: chất lượng dịch vụ, độ tin cậy, khả năng vận chuyển hoặc an ninh. Mô hình end-to-end đẩy độ phức tạp lên các máy đầu cuối, để mạng trung gian trở nên đơn giản và được tối ưu hóa cho việc chuyển tiếp các gói tin. Nếu một bên trong giao tiếp hai chiều (người gửi hoặc người nhận) hoạt động không đúng theo thiết kế thì có thể gây thiệt hại tùy ý cho đồng nghiệp của mình. Mạng trung gian sẽ không ngăn chặn điều này, bởi vì Internet không được thiết kế để giám sát quá trình truyền thông. Một hệ quả của chính sách này là sự hiện diện của IP giả mạo. Việc có thể giả mạo địa chỉ IP cho phép những kẻ tấn công một cơ chế mạnh mẽ để thoát khỏi trách nhiệm về hành động của họ và đôi khi thậm chí là phương tiện để thực hiện các cuộc tấn công.

DDoS có thể được thực hiện bởi nhiều nguyên nhân khác nhau, bao gồm:

- Botnet: Kẻ tấn công sử dụng botnet để thực hiện tấn công. Botnet là một mạng các thiết bị bị lây nhiễm bởi phần mềm độc hại và được điều khiển từ xa bởi tin tặc.
- Máy tính bị nhiễm virus: Máy tính của người dùng cuối cũng có thể bị lây nhiễm bởi phần mềm độc hại, biến chúng thành một phần của botnet và thực hiện cuộc tấn công.
- Tài nguyên hạn chế: Các trang web hoặc dịch vụ web có tài nguyên hạn chế hoặc không đủ khả năng để chống lại cuộc tấn công DDoS.
- Lỗ hổng bảo mật: Các lỗ hổng bảo mật trong phần mềm hoặc hệ thống mạng cũng có thể được khai thác để thực hiện cuộc tấn công DDoS.
- Sử dụng công cụ tấn công DDoS: Một số kẻ tấn công có thể sử dụng các công cụ tấn công DDoS có sẵn để thực hiện các cuộc tấn công mà không cần có kiến thức chuyên môn về kỹ thuật.

1.2. Các giai đoạn của một cuộc tấn công DDoS

Một cuộc tấn công DDoS thường bao gồm các giai đoạn sau:

- Thu thập thông tin: Kẻ tấn công thường sẽ tiến hành thu thập thông tin về mục tiêu của mình, bao gồm địa chỉ IP, cổng mạng, loại hệ thống và ứng dụng đang chạy.

- Tìm kiếm lỗ hổng: Sau khi có thông tin về mục tiêu, kẻ tấn công sẽ thực hiện các cuộc kiểm tra để tìm kiếm lỗ hổng bảo mật trong hệ thống. Khi tìm thấy lỗ hổng, họ sẽ sử dụng các công cụ và kỹ thuật để khai thác và tận dụng lỗ hổng đó.
- Xâm nhập: Sau khi khai thác thành công lỗ hổng bảo mật, kẻ tấn công sẽ xâm nhập vào hệ thống của mục tiêu và cài đặt phần mềm độc hại để kiểm soát các thiết bị.
- Lây lan: Khi đã có quyền kiểm soát các thiết bị trong hệ thống, kẻ tấn công sẽ sử dụng chúng để tấn công các thiết bị khác trên mạng, tạo thành một mạng botnet.
- Phát tán: Kẻ tấn công sẽ sử dụng mạng botnet để phát tán các yêu cầu đến mục tiêu, gây quá tải hệ thống và làm cho hệ thống không thể hoạt động bình thường.
- Giấu dấu vết: Sau khi tấn công, kẻ tấn công sẽ xoá các tài khoản, các bản ghi lịch sử truy cập, và các dấu vết khác để giấu mình.

1.3. Phân loại tấn công từ chối dịch vụ phân tán(DDoS)

Các loại tấn công DDoS có rất nhiều dạng và biến thể khác nhau, dẫn đến cũng sẽ có rất nhiều cách phân loại khác nhau. Tuy nhiên, trong bài báo cáo này chúng tôi phân lại ra làm 4 loại chính:

- Volume-based attacks: là một loại tấn công DDoS nhắm vào tối đa hóa tài nguyên mạng bằng cách tăng lưu lượng truy cập đến một máy chủ hoặc một mạng bằng cách tạo ra một lượng lớn yêu cầu từ các máy tính của botnet, cường độ được đo bằng bit trên giây (Bps). Các tấn công Volume-based attacks được xếp vào nhóm tấn công "tấn công cực lớn" (High-rate attack). Tấn công Volume-based attacks có thể gây ra thiệt hại đáng kể cho các tổ chức, doanh nghiệp, các trang web, trò chơi trực tuyến và hệ thống mạng, khiến chúng không thể hoạt động được bình thường, gây mất cân bằng tài nguyên và ảnh hưởng đến trải nghiệm người dùng. Một số loại tấn công DDoS thể hiện lưu lượng cao bao gồm: ICMP flood, UDP flood, và SYN flood.
- Protocol attacks: Là một trong các loại tấn công DDoS nhằm vào lỗ hổng của các giao thức mạng, ví dụ như TCP, UDP, DNS, ICMP, và SNMP, để gây tắc nghẽn và làm cho các hệ thống mạng hoạt động chậm hơn hoặc tạm thời ngừng hoạt động, được đo bằng gói mỗi giây (Pps). Các tấn công này thường liên quan đến việc gửi một lượng lớn các gói tin giả mạo hoặc có nội dung không hợp lệ tới các máy chủ mạng. Ví dụ, tấn công SYN Flood tấn công vào giao thức TCP bằng cách gửi hàng loạt các yêu cầu kết nối TCP giả mạo tới máy chủ, gây tắc nghẽn và làm cho máy chủ không thể phục vụ được các yêu cầu hợp lệ. Các tấn công Protocol attacks có thể gây ra thiệt hại nghiêm trọng đối với các hệ thống mạng và ảnh hưởng đến sự hoạt động của các dịch vụ trực tuyến, ví dụ như các trang web, hệ thống ngân hàng trực tuyến, và các ứng dụng truyền thông. Một số loại tấn công giao thức như: SYN floods, tấn công gói phân mảnh, Ping of Death, Smurf DDoS.

- Application Layer Attacks: Tấn công này nhằm tấn công vào các ứng dụng, đặc biệt là các ứng dụng web hoặc ứng dụng di động của mục tiêu. Được tạo thành từ các yêu cầu có vẻ hợp lệ và vô hại, mục tiêu của những cuộc tấn công này là làm sập máy chủ web, và quy mô được đo bằng số yêu cầu mỗi giây (Rps). Điều này khác với các tấn công tầng khác vì chúng tập trung vào mục tiêu cụ thể hơn thay vì chỉ đơn thuần tấn công đến hệ thống. Các cuộc tấn công ở tầng ứng dụng này thường khó phát hiện và khó khắc phục, đòi hỏi những giải pháp bảo mật phức tạp hơn để ngăn chặn và giảm thiểu tác động của chúng. Loại tấn công này bao gồm các cuộc tấn công low-and-slow, GET/POST floods, các cuộc tấn công nhắm vào các lỗ hổng của Apache, Windows hoặc OpenBSD.
- Tấn công DDoS tiêu diệt máy chủ (Permanent Denial-of-Service attack): là một loại tấn công mạng nhằm hủy hoại tới cơ sở hạ tầng của hệ thống. Các tấn công PDoS cố gắng làm hỏng các thiết bị, phần mềm hoặc tài nguyên mạng của một hệ thống, từ đó khiến cho hệ thống không thể hoạt động được nữa. Tương tự như các loại tấn công khác, PDoS được thực hiện bằng cách gửi một lượng lớn yêu cầu đến một phần của hoặc toàn bộ hệ thống, vượt quá khả năng xử lý của nó. Tuy nhiên, điểm khác biệt của PDoS so với các loại tấn công DDoS khác là chúng cố gắng hủy hoại thiết bị và phần mềm bằng cách tận dụng các lỗ hổng bảo mật. Tấn công PDoS là một dạng tấn công nguy hiểm, có thể dẫn đến thiệt hại vĩnh viễn cho hệ thống bị tấn công.

1.4. Những hình thức tấn công DDoS thường gặp phải

1.4.1. SYN Flood

SYN Flood là một loại tấn công DDoS thuộc nhóm tấn công volumetric, tập trung vào việc làm cho một trang web hoặc một dịch vụ mạng không thể sử dụng được bằng cách tràn bộ đệm của máy chủ đích với các gói tin SYN (synchronization) giả mạo.

Trong một kết nối TCP, máy khách gửi một gói tin SYN đến máy chủ và yêu cầu một kết nối mới. Sau đó, máy chủ phản hồi bằng một gói tin SYN-ACK để xác nhận yêu cầu kết nối và đợi máy khách gửi một gói tin ACK để hoàn tất quá trình kết nối. Tuy nhiên, trong cuộc tấn công SYN Flood, kẻ tấn công gửi nhiều gói tin SYN giả mạo đến máy chủ, khiến cho máy chủ không thể phản hồi chính xác với tất cả các yêu cầu này. Kết quả là, máy chủ sẽ lưu giữ và tiếp nhận các kết nối giả mạo này trong một khoảng thời gian nhất định, chiếm dụng tài nguyên hệ thống và làm cho dịch vụ trở nên không sử dụng được.

Ba cách mà một cuộc tấn công SYN Flood có thể xảy ra như sau:

- Tấn công trực tiếp là khi kẻ tấn công không giả mạo địa chỉ IP và sử dụng một thiết bị nguồn duy nhất với địa chỉ IP thực để thực hiện tấn công SYN. Phương pháp này dễ dàng bị phát hiện và giảm thiểu hiệu quả của cuộc tấn công.
- Tấn công giả mạo: Để ngăn chặn những nỗ lực giảm thiểu và che dấu danh tính, kẻ tấn công sẽ giả mạo địa chỉ IP trên các gói packet SYN mà chúng gửi tới

server. Để có thể truy ngược trở lại nguồn gốc của các gói tin không phải là điều dễ dàng. Tuy nhiên nếu có sự giúp đỡ của các nhà cung cấp dịch vụ Internet thì hoàn toàn có thể thực hiện được.

- Tấn công phân tán: Khả năng thành công của cách tấn công phân tán DDoS rất cao bởi nó được tạo ra bằng mạng botnet và để theo dõi nguồn gốc là điều khá khó khăn. Với sự xáo trộn và che dấu được thêm vào, kẻ tấn công sẽ có các thiết bị phân tán đồng thời giả mạo các địa chỉ IP trên các gói packet.

Khác với các hình thức tấn công DDoS khác, tấn công DDoS SYN Flood không nhằm mục đích sử dụng hết tài nguyên bộ nhớ của máy chủ. Thay vào đó, tấn công này dùng để tiêu tốn tài nguyên dự trữ của các kết nối mở thông qua cổng từ các địa chỉ IP riêng lẻ, thường là giả mạo. Tấn công SYN Flood được gọi là "công khai nửa" bởi vì nó gửi một loạt các tin nhắn SYN ngắn đến các cổng, mở các kết nối không an toàn và dẫn đến sự cố hoàn toàn của máy chủ.

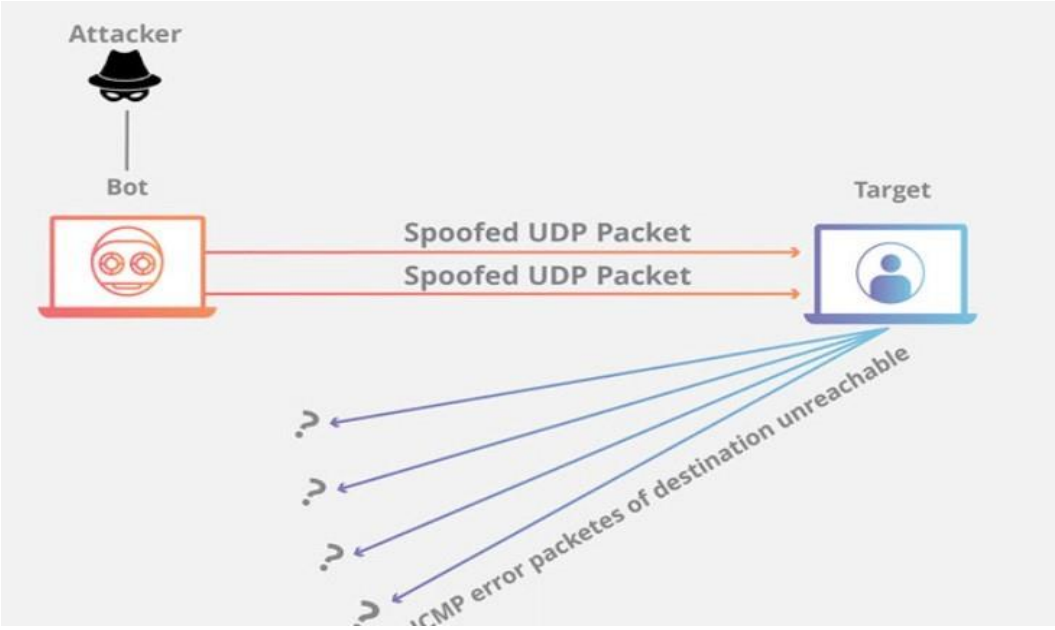
SYN Flood là một loại tấn công phổ biến vì nó đơn giản và dễ thực hiện. Kẻ tấn công chỉ cần tạo ra và gửi một số lượng lớn các gói tin SYN giả mạo đến máy chủ đích từ nhiều địa chỉ IP khác nhau. Để ngăn chặn tấn công SYN Flood, các biện pháp bảo vệ phải được triển khai như sử dụng tường lửa và load balancer hoặc các công cụ phát hiện tấn công DDoS để xác định và chặn các yêu cầu SYN giả mạo.

Một số cách để giảm thiểu các cuộc tấn công lũ lụt SYN, bao gồm các kỹ thuật sau:

- Tăng hàng đợi Backlog: Hệ điều hành của thiết bị mà kẻ xấu nhắm vào sẽ có một số kết nối half – open cho phép. Tăng số lượng kết nối half – open là một cách hay để giảm thiểu tấn công SYN. Và để tăng backlog tồn đọng thì hệ thống phải dự trữ thêm bộ nhớ. Nếu bộ nhớ không đủ để xử lý backlog tồn đọng thì hiệu suất làm việc của hệ thống bị ảnh hưởng.
- Tái chế kết nối TCP Half-Open cũ: Sau khi backlog được lấp đầy thì lặp lại kết nối half – open TCP cũ là một chiến lược để giảm thiểu cuộc tấn công SYN. Cách thức này yêu cầu trong một thời gian ngắn các kết nối hợp pháp được thiết lập thay vì backlog sẽ chứa nhiều gói SYN độc hại. Tuy nhiên phương án này sẽ thất bại khi các cuộc tấn công trở nên mạnh mẽ hoặc kích thước backlog quá nhỏ.
- SYN cookie: Tạo ra một cookie của server là một chiến lược hay ho để hạn chế sự tấn công lũ lượt của SYN. Server sẽ dùng gói packet SYN – ACK để phản hồi từng kết nối và xóa yêu cầu SYN ra khỏi backlog, để port mở sẵn sàng tạo kết nối mới. Hành động này nhằm mục đích tránh rủi ro rớt kết nối khi mà backlog đã được lấp đầy.

1.4.2. UDP Flood

UDP Flood là một loại tấn công từ chối dịch vụ (DDoS) được thực hiện bằng cách gửi một lượng lớn các gói tin UDP (User Datagram Protocol) không hợp lệ đến một địa chỉ IP đích. Đây là một loại tấn công thường được sử dụng để làm quá tải và làm ngưng hoạt động các ứng dụng trên mạng, ví dụ như các trang web, ứng dụng trò chơi trực tuyến và các dịch vụ truyền phát video.



Hình 2 : Tấn công UDP Flood

UDP Flood là một cuộc tấn công khai thác các bước mà máy chủ thực hiện khi nó nhận được gói UDP được gửi đến một trong các cổng của nó. Thông thường, khi nhận được gói UDP tại một cổng cụ thể, máy chủ sẽ thực hiện hai bước sau đây để phản hồi:

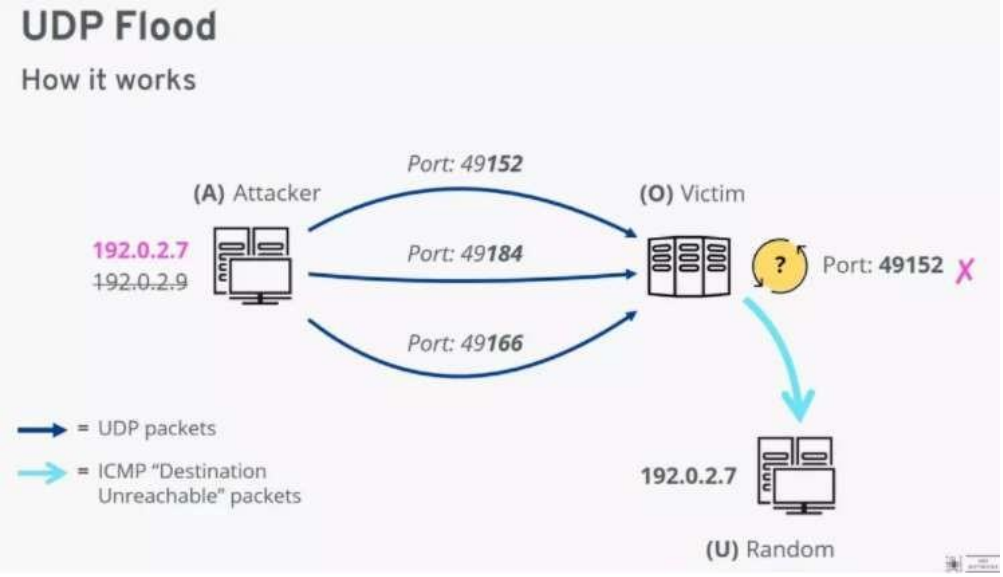
- Đầu tiên, máy chủ sẽ kiểm tra xem có chương trình nào đang chạy đang lắng nghe yêu cầu tại cổng được chỉ định hay không.
- Nếu không có chương trình nào nhận gói tại cổng đó, máy chủ sẽ phản hồi bằng gói ICMP (ping) để thông báo cho người gửi rằng không thể truy cập đích

Khi máy chủ nhận được mỗi gói UDP mới, nó sẽ trải qua các bước để xử lý yêu cầu, sử dụng tài nguyên máy chủ trong quy trình. Khi các gói UDP được truyền đi, mỗi gói sẽ bao gồm địa chỉ IP của thiết bị nguồn. Trong kiểu tấn công DDoS này, kẻ tấn công thường sẽ không sử dụng địa chỉ IP thực của chính chúng mà thay vào đó sẽ giả mạo địa chỉ IP nguồn của các gói UDP, ngăn không cho vị trí thực của kẻ tấn công bị lộ và có khả năng bị bão hòa với các gói phản hồi từ mục tiêu máy chủ.

Do máy chủ được nhắm mục tiêu sử dụng tài nguyên để kiểm tra và sau đó phản hồi từng gói UDP nhận được, tài nguyên của mục tiêu có thể nhanh chóng cạn kiệt khi

nhận được một lượng lớn gói UDP, dẫn đến từ chối dịch vụ đối với lưu lượng truy cập bình thường.

Là một cuộc tấn công từ chối dịch vụ phổ biến, lũ lụt UDP có thể dễ dàng khiến máy chủ hoặc ứng dụng không khả dụng cho người dùng. Điều này có thể nhanh chóng dẫn đến sự sụt giảm đáng kể về năng suất, mất doanh thu, tổn hại đến danh tiếng và sự rời bỏ của khách hàng. Các cuộc tấn công tràn ngập UDP được coi là đặc biệt nguy hiểm vì không có biện pháp bảo vệ nội bộ nào có thể hạn chế tốc độ tràn ngập UDP, vì vậy chúng có thể được thực hiện bởi những kẻ tấn công với rất ít tài nguyên.



Hình 3: Quá trình tấn công UDP Flood Attack

Flood UDP nguy hiểm hơn Flood TCP vì UDP là một giao thức không kết nối. Điều này có nghĩa là không cần thiết lập kết nối trước khi gửi dữ liệu. Flood UDP có thể dễ dàng áp đảo máy chủ bằng các gói giả mạo.

Các kỹ thuật phòng ngừa tấn công UDP Flood bao gồm giới hạn số lượng gói tin UDP được chấp nhận từ một địa chỉ IP cụ thể, sử dụng các giải pháp bảo mật mạng như bộ tường lửa và phát hiện tấn công từ chối dịch vụ (DDoS) tự động để phát hiện và chặn các cuộc tấn công này.

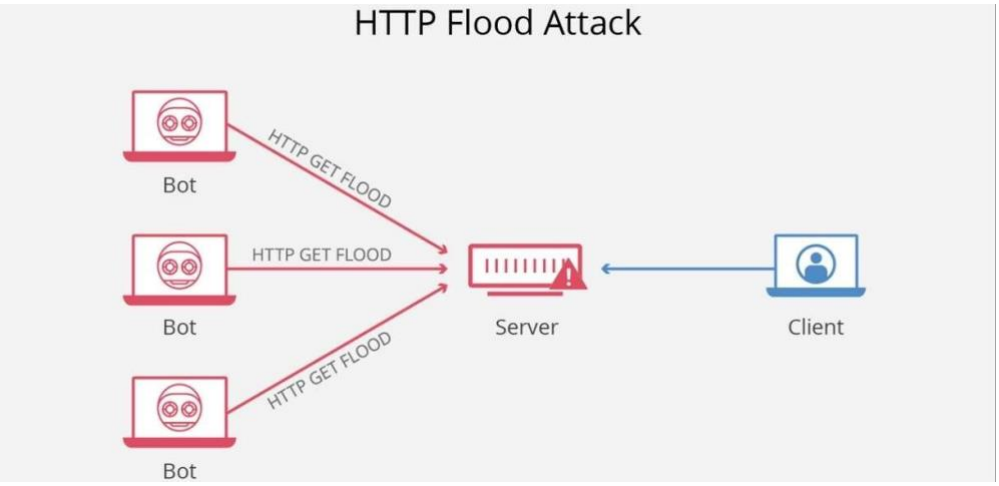
Một số cách chống UDP Flood như:

- Thiết lập giới hạn tốc độ truyền dữ liệu đến các cổng UDP: Sử dụng các công cụ như "iptables" hoặc "irewall" để giới hạn tốc độ truyền dữ liệu đến các cổng UDP.
- Sử dụng các giải pháp cân bằng tải: Sử dụng các giải pháp cân bằng tải để phân phối tải cho các máy chủ khác nhau, giảm thiểu khả năng tấn công.

- Sử dụng các dịch vụ bảo vệ chống tấn công DDoS: Có nhiều dịch vụ bảo vệ chống tấn công DDoS có thể giúp giảm thiểu tác động của UDP Flood Attack bằng cách lọc lưu lượng truy cập đến server.
- Tối ưu hóa cấu hình máy chủ: Tối ưu hóa cấu hình máy chủ để giảm thiểu tác động của UDP Flood Attack. Ví dụ như tăng dung lượng bộ nhớ đệm (buffer) cho giao thức UDP.

1.4.3. HTTP Flood

HTTP Flood là một trong những loại tấn công DDoS phổ biến nhất, nó tấn công vào tầng ứng dụng của một trang web bằng cách gửi nhiều yêu cầu HTTP đến máy chủ Web. Mục đích của tấn công là làm quá tải máy chủ và làm cho trang Web trở nên không khả dụng cho người dùng bình thường.



Hình 4: Tấn công HTTP Flood

Các cuộc tấn công HTTP flood là một loại tấn công ở tầng 7 (layer 7) của mô hình OSI, nghĩa là tấn công vào các giao thức Internet như HTTP. HTTP là giao thức cơ bản được sử dụng để truyền tải các yêu cầu của trình duyệt web. Cuộc tấn công HTTP flood đặc biệt phức tạp để giảm thiểu, bởi vì lưu lượng tấn công rất khó phân biệt với lưu lượng thông thường.

Tấn công HTTP Flood thường được thực hiện bằng cách sử dụng botnet, một mạng lưới các máy tính được kiểm soát từ xa bởi kẻ tấn công. Các bot trong botnet sẽ gửi hàng nghìn, hàng triệu yêu cầu HTTP đến máy chủ Web cùng lúc. Điều này gây ra một số vấn đề cho máy chủ Web, bao gồm khả năng xử lý yêu cầu và lưu trữ dữ liệu, dẫn đến việc trang web trở nên không khả dụng.

Có hai loại tấn công HTTP flood, đó là:

- Tấn công HTTP GET: Đây là hình thức tấn công mà nhiều máy tính hoặc thiết bị khác nhau phối hợp để gửi nhiều yêu cầu đến server mục tiêu, yêu cầu này có thể là hình ảnh, tệp tin hoặc dữ liệu khác. Khi server mục tiêu không thể xử lý được

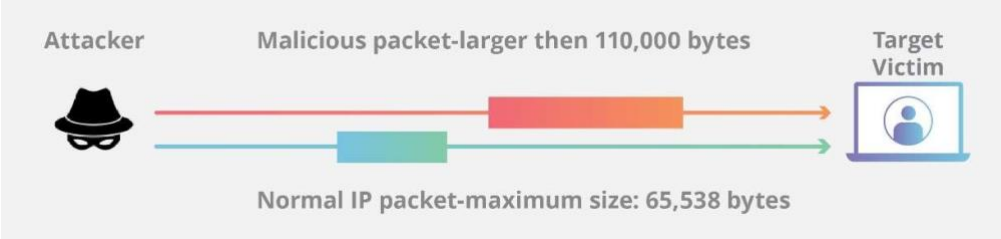
số lượng yêu cầu đó và trả về phản hồi, các yêu cầu bổ sung từ các nguồn lưu lượng hợp pháp sẽ bị từ chối, dẫn đến tình trạng từ chối dịch vụ.

– Tấn công POST HTTP: Thông thường, khi một biểu mẫu được gửi từ trang web, server phải xử lý yêu cầu và lưu dữ liệu vào một lớp liên tục, thường là cơ sở dữ liệu. Quá trình xử lý dữ liệu biểu mẫu và thực hiện các lệnh cơ sở dữ liệu là rất tốn kém về sức mạnh xử lý và băng thông. Cuộc tấn công này sử dụng sự chênh lệch về mức tiêu thụ tài nguyên, bằng cách gửi trực tiếp nhiều yêu cầu bài đến server được nhắm mục tiêu cho đến khi dung lượng của nó bị bão hòa và xảy ra sự từ chối dịch vụ.

Việc giảm thiểu các cuộc tấn công layer 7 là rất phức tạp và cần nhiều phương pháp. Một trong số đó là thực hiện thách thức cho máy yêu cầu để kiểm tra xem đó có phải là bot hay không, giống như thử nghiệm captcha thường được sử dụng để tạo tài khoản trực tuyến. Bằng cách đưa ra một yêu cầu thách thức tính toán JavaScript, nhiều cuộc tấn công có thể được giảm thiểu. Các phương pháp khác để ngăn chặn HTTP flood bao gồm sử dụng tường lửa ứng dụng web (WAF), quản lý cơ sở dữ liệu địa chỉ IP uy tín để theo dõi và chặn các lưu lượng độc hại.

1.4.4. Ping of Death

Ping of Death là một dạng tấn công mạng sử dụng gói tin ping (ICMP), trong đó kẻ tấn công gửi các gói tin ping chứa kích thước lớn hơn giới hạn cho phép, gây ra tràn bộ đệm và tạo ra lỗi hệ thống, và kiểu tấn công này sẽ thường bắt gặp trên các hệ điều hành Windows NT trở xuống. Tấn công Ping of Death thường được thực hiện bằng cách gửi các gói tin ICMP có kích thước lớn hơn 65.535 byte, giới hạn cho phép trong giao thức ICMP. Khi một gói lớn độc hại được truyền từ kẻ tấn công đến mục tiêu, gói sẽ bị phân mảnh thành các phân đoạn, mỗi phân đoạn nhỏ hơn giới hạn kích thước tối đa. Khi máy mục tiêu cố gắng ghép các phần lại với nhau, tổng số vượt quá giới hạn kích thước và có thể xảy ra lỗi tràn bộ đệm, khiến máy mục tiêu bị đóng băng, gặp sự cố hoặc khởi động lại.



Hình 5: Mô tả tấn công Ping of Death

Các cuộc tấn công Ping of Death đặc biệt hiệu quả, vì danh tính của kẻ tấn công có thể dễ dàng giả mạo. Hơn nữa, kẻ tấn công Ping of Death sẽ không cần tìm hiểu quá chi tiết về máy mà hắn ta đang tấn công, ngoại trừ địa chỉ IP của nó.

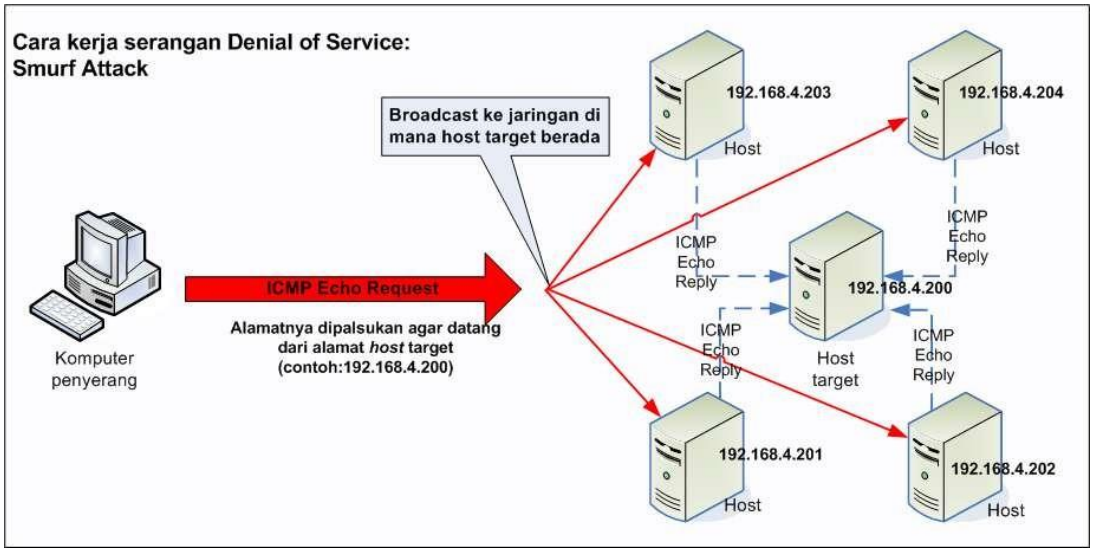
Cần lưu ý rằng lỗ hổng này, mặc dù được công nhận là có khả năng khai thác tốt nhất bởi các cuộc tấn công PoD, thực sự có thể bị khai thác bởi bất kỳ thứ gì gửi một IP datagram - ICMP echo, TCP, UDP và IPX.

Tấn công Ping of Death không phổ biến nhưng vẫn được sử dụng trong một số trường hợp. Tấn công DDOS kiểu Ping of Death này phổ biến ở 2 thập kỷ trước hơn là hiện tại, cho nên thường không mang lại hiệu quả cao ở thời điểm này. Hầu hết các hệ thống đều có cơ chế bảo vệ để ngăn chặn tấn công này bằng cách giới hạn kích thước gói tin ICMP hoặc sử dụng các bộ lọc bảo vệ đường truyền.

Hầu hết các mạng vận hành tường lửa cho phép các tổ chức chặn thông báo ping ICMP. Điều này sẽ cho phép họ chặn các cuộc tấn công ping of death nhưng không phải là một cách tiếp cận thực tế vì nó ảnh hưởng đến hiệu suất và độ tin cậy cũng như chặn các ping hợp pháp. Chúng cũng không lý tưởng—các cuộc tấn công gói không hợp lệ có thể được khởi chạy thông qua các cổng lắng nghe như Giao thức truyền tệp (FTP).

1.4.5. Smurf Attack

Smurf Attack là một loại tấn công DDoS mà kẻ tấn công gửi các gói tin ping đến các địa chỉ IP nguồn giả mạo, nhằm khiến các máy chủ trung gian (intermediate servers) hoặc các thiết bị mạng phản hồi bằng gửi các gói tin ping đáp lại với địa chỉ IP đích là máy chủ mà kẻ tấn công muốn tấn công. Bằng cách gửi các gói tin ping với kích thước lớn đến các địa chỉ IP nguồn giả mạo, kẻ tấn công có thể gây ra một lượng lớn dữ liệu liên quan đến các yêu cầu ping, tạo ra một lưu lượng mạng lớn và khiến cho máy chủ đích không thể xử lý được lưu lượng này, từ đó dẫn đến việc bị tắt mạng hoặc bị chậm trễ. Các cuộc tấn công Smurf hơi giống với ping floods, vì cả hai đều được thực hiện bằng cách gửi một loạt các gói yêu cầu ICMP Echo. Tuy nhiên, không giống như ping floods thông thường, Smurf là một véc tơ tấn công khuếch đại giúp tăng khả năng sát thương của nó bằng cách khai thác các đặc điểm của mạng phát sóng.



Hình 6: Mô tả Smurf Attack

Có thể vô tình tải xuống Trojan Smurf từ một trang web chưa được xác minh hoặc qua liên kết email bị nhiễm. Thông thường, chương trình sẽ không hoạt động trên máy tính cho đến khi được kích hoạt bởi người dùng từ xa; kết quả là nhiều Smurf đi kèm với rootkit, cho phép tin tặc tạo các cửa hậu để truy cập hệ thống dễ dàng. Một cách để chống lại cuộc tấn công Smurf là tắt địa chỉ quảng bá IP trên mọi bộ định tuyến mạng. Chức năng này hiếm khi được sử dụng và nếu bị tắt, cuộc tấn công sẽ không thể áp đảo mạng.

Nếu một cuộc tấn công DDoS của Smurf thành công, nó có thể làm tê liệt các máy chủ của công ty trong nhiều giờ hoặc nhiều ngày, dẫn đến mất doanh thu và sự thất vọng của khách hàng — hơn nữa, kiểu tấn công này cũng có thể là vỏ bọc cho một thứ gì đó nguy hiểm hơn, chẳng hạn như đánh cắp tệp hoặc tài sản trí tuệ (IP) khác.

Cách smurf attack hoạt động:

- Phần mềm độc hại Smurf được sử dụng để tạo yêu cầu Echo giả có chứa IP nguồn giả mạo, đây thực sự là địa chỉ máy chủ đích.
- Yêu cầu được gửi đến một mạng quảng bá IP trung gian.
- Yêu cầu được truyền đến tất cả các máy chủ mạng trên mạng.
- Mỗi máy chủ gửi phản hồi ICMP đến địa chỉ nguồn giả mạo.
- Với đủ phản hồi ICMP được chuyển tiếp, máy chủ mục tiêu sẽ ngừng hoạt động.

Hệ số khuếch đại của cuộc tấn công Smurf tương quan với số lượng máy chủ trên mạng trung gian. Ví dụ: mạng quảng bá IP có 500 máy chủ sẽ tạo ra 500 phản hồi cho mỗi yêu cầu Echo giả. Thông thường, mỗi phần phụ thuộc có cùng kích thước với yêu cầu ping ban đầu.

Smurf Attack đã được coi là một trong những cuộc tấn công DDoS đáng sợ nhất vào những năm 1990 và đầu 2000, tuy nhiên hiện nay đã ít được sử dụng hơn do các cơ quan bảo mật đã nâng cao kiến thức và các biện pháp phòng ngừa. Smurf Attack có thể được ngăn chặn bằng cách sử dụng các bộ lọc địa chỉ IP để chặn các gói tin ping với địa chỉ nguồn giả mạo trước khi chúng được gửi đến mạng. Bên cạnh đó, các máy chủ trung gian cũng nên được cấu hình để không trả lời các gói tin ping từ các địa chỉ nguồn không hợp lệ.

1.4.6. Fraggle Attack

Fraggle Attack là một hình thức tấn công DDoS giống với Smurf Attack, tuy nhiên nó sử dụng giao thức UDP thay vì ICMP. Kẻ tấn công sử dụng địa chỉ IP giả mạo để gửi các yêu cầu phản hồi từ máy chủ UDP. Tương tự như Smurf Attack, các phản hồi này được gửi đến một địa chỉ IP nhắm mục tiêu và được tạo ra như một gói tin lớn, đánh cắp lưu lượng mạng của mục tiêu và gây ra sự cố mạng.

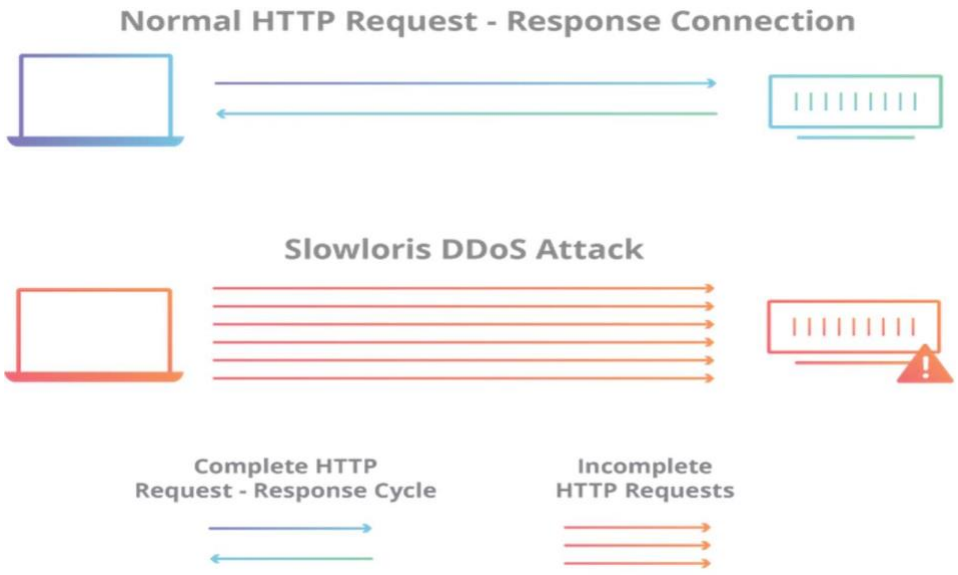
Fraggle Attack cũng có thể được chủ động hoặc bị lợi dụng từ một máy tính đã bị tống khứ từ mạng (zombie hoặc botnet), khiến nó trở thành một công cụ trong tay kẻ

tấn công. Tương tự như Smurf Attack, Fraggle Attack cũng đã được phát hiện và có các biện pháp phòng ngừa tương tự.

1.4.7. Slowloris

Slowloris là một kiểu tấn công từ chối dịch vụ (DDoS) tập trung vào ứng dụng web. Nó hoạt động bằng cách mở nhiều kết nối đến máy chủ web đích và duy trì các kết nối này một cách liên tục, đồng thời không hoàn thành các yêu cầu HTTP đầy đủ. Máy chủ được nhắm mục tiêu sẽ chỉ có rất ít luồng có sẵn để xử lý các kết nối đồng thời. Mỗi luồng máy chủ sẽ cố gắng duy trì sự sống trong khi chờ yêu cầu chậm hoàn thành, điều này không bao giờ xảy ra. Khi vượt quá các kết nối tối đa có thể của máy chủ, mỗi kết nối bổ sung sẽ không được trả lời và từ chối dịch vụ sẽ xảy ra.

Cách thức tấn công Slowloris khác với các loại tấn công khác như SYN Flood hay UDP Flood. Trong khi các tấn công đó sử dụng nhiều gói tin để gửi đến máy chủ đích trong thời gian ngắn, Slowloris duy trì các kết nối dài hạn và không đầy đủ, làm cho việc phát hiện và chặn tấn công này khó khăn hơn.



Hình 7: Mô tả tấn công Slowloris

Cuộc tấn công Slowloris xảy ra theo 4 bước như sau:

- Kẻ tấn công mở nhiều kết nối đến máy chủ bằng cách gửi nhiều yêu cầu HTTP.
- Mục tiêu sẽ mở một luồng cho mỗi yêu cầu đến và đóng luồng sau khi kết nối hoàn tất. Nếu kết nối mất quá nhiều thời gian, máy chủ sẽ hết thời gian kết nối quá dài và giải phóng chuỗi cho yêu cầu tiếp theo.
- Kẻ tấn công định kỳ gửi các tiêu đề yêu cầu một phần đến mục tiêu để giữ cho yêu cầu tồn tại và ngăn chặn mục tiêu hết thời gian kết nối.
- Máy chủ được nhắm mục tiêu sẽ không bao giờ phát hành bất kỳ kết nối một phần mở nào trong khi đợi kết thúc yêu cầu. Khi tất cả các luồng đang được sử

dụng, máy chủ sẽ không thể đáp ứng các yêu cầu bổ sung được thực hiện từ lưu lượng truy cập thông thường, gây ra tình trạng từ chối dịch vụ.

Một trong những đặc điểm nổi bật của Slowloris là nó có thể được thực hiện từ một máy tính cá nhân đơn lẻ, chỉ cần một kết nối internet đủ mạnh để tạo ra các kết nối TCP đến máy chủ web đích. Do đó, đây là một trong những loại tấn công DDoS có thể được thực hiện bởi các tấn công viên cá nhân và không yêu cầu nhiều tài nguyên hoặc một mạng botnet lớn.

1.4.8. NTP Amplification

Tấn công Amplification NTP là một loại tấn công DDoS tận dụng Giao thức Thời gian Mạng (NTP), được sử dụng để đồng bộ hóa đồng hồ máy tính qua mạng. Trong tấn công này, kẻ tấn công gửi một số lượng nhỏ các gói NTP đến một máy chủ dễ bị tổn thương với địa chỉ IP nguồn giả mạo. Sau đó, máy chủ phản hồi với số lượng gói lớn hơn đáp trả nạn nhân, tăng cường lưu lượng và áp đảo mạng của họ. Với kỹ thuật này, kẻ tấn công có thể tạo ra một lượng lớn lưu lượng mạng và gây ra tình trạng quá tải cho hệ thống mạng đích, làm gián đoạn hoạt động của nó hoặc khiến nó không thể truy cập được từ bên ngoài.

Cuộc tấn công này đặc biệt hiệu quả vì các máy chủ NTP có thể tạo ra các phản hồi lớn hơn tới 1000 lần so với yêu cầu ban đầu, giúp tạo ra một lượng lớn lưu lượng truy cập với số lượng gói tương đối nhỏ. Các cuộc tấn công khuếch đại NTP là nguyên nhân gây ra một số cuộc tấn công DDoS lớn nhất trong lịch sử và chúng vẫn là mối đe dọa nghiêm trọng đối với các mạng trên toàn thế giới.

Để giảm thiểu các cuộc tấn công khuếch đại NTP, nên tắt NTP monlist và hạn chế quyền truy cập vào các dịch vụ NTP chỉ cho các máy khách đáng tin cậy. Quản trị viên mạng cũng có thể triển khai các quy tắc tường lửa hoặc giới hạn tốc độ để giới hạn lượng lưu lượng NTP có thể được gửi đến một địa chỉ IP cụ thể.

1.4.9. Advanced persistent Dos (APDos)

Advanced Persistent DoS (APDoS) là một loại tấn công DDoS có tính toàn diện, chủ yếu nhằm vào các mục tiêu quan trọng như các tổ chức chính phủ, các cơ quan tình báo, các công ty tài chính hoặc các tổ chức lớn. Loại tấn công này là sự kết hợp của nhiều kỹ thuật tấn công khác nhau, chủ yếu là tấn công lớp 3 và lớp 4, và được thực hiện trong một thời gian dài, thường là nhiều tháng hoặc nhiều năm.

APDoS thường sử dụng các kỹ thuật giả mạo địa chỉ IP, kết hợp với các công cụ và phần mềm tấn công chuyên nghiệp để tạo ra lưu lượng truy cập lớn đến các mục tiêu, gây ra tình trạng quá tải và làm cho dịch vụ của các mục tiêu không thể hoạt động được. APDoS có thể gây thiệt hại nghiêm trọng cho các tổ chức bị tấn công, bao gồm mất dữ liệu, mất tài nguyên, gián đoạn hoạt động và giảm uy tín của tổ chức.

CHƯƠNG 2: TỔNG QUAN VỀ PHÂN LỚP DỮ LIỆU

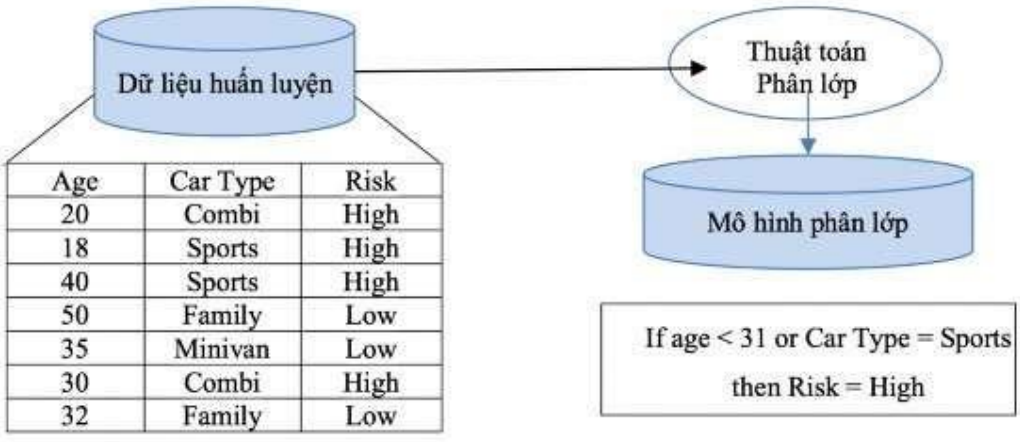
2.1. Giới thiệu về phân lớp dữ liệu

Phân loại dữ liệu được định nghĩa rộng rãi là quá trình tổ chức dữ liệu theo các danh mục có liên quan để dữ liệu có thể được sử dụng và bảo vệ hiệu quả hơn. Ở cấp độ cơ bản, quy trình phân loại giúp định vị và truy xuất dữ liệu dễ dàng hơn. Phân loại dữ liệu có tầm quan trọng đặc biệt khi nói đến quản lý rủi ro, tuân thủ và bảo mật dữ liệu.

Phân loại dữ liệu liên quan đến việc gắn thẻ dữ liệu để dễ dàng tìm kiếm và theo dõi. Nó cũng loại bỏ nhiều dữ liệu trùng lặp, có thể giảm chi phí lưu trữ và sao lưu đồng thời tăng tốc quá trình tìm kiếm.

Quá trình phân loại dữ liệu bao gồm hai bước chính:

- Bước thứ nhất (học tập): Quá trình học tập nhằm xây dựng một mô hình để miêu tả một tập các lớp dữ liệu hoặc các khái niệm đã được xác định trước. Đầu vào của quá trình này là một tập dữ liệu có cấu trúc, được miêu tả bởi các thuộc tính và được tạo ra từ các bộ giá trị của các thuộc tính đó. Mỗi bộ giá trị được gọi là một phần tử dữ liệu, có thể là các mẫu, ví dụ, đối tượng, bản ghi hoặc trường hợp. Luận án sử dụng các thuật ngữ này với nghĩa tương đương. Trong tập dữ liệu này, mỗi phần tử dữ liệu được giả định rằng thuộc về một lớp được xác định trước, trong đó lớp là giá trị của một thuộc tính được chọn làm thuộc tính gán nhãn lớp hoặc thuộc tính phân loại. Kết quả đầu ra của bước này thường là các quy tắc phân loại dưới dạng luật if-then, cây quyết định, công thức logic hoặc mạng nơron.



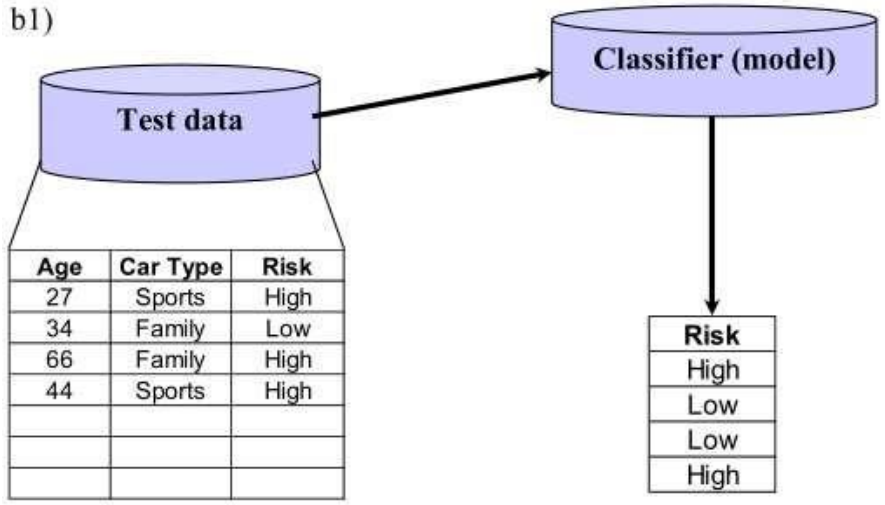
Age	Car Type	Risk
20	Combi	High
18	Sports	High
40	Sports	High
50	Family	Low
35	Minivan	Low
30	Combi	High
32	Family	Low

Hình 8: Bước xây dựng mô hình phân lớp

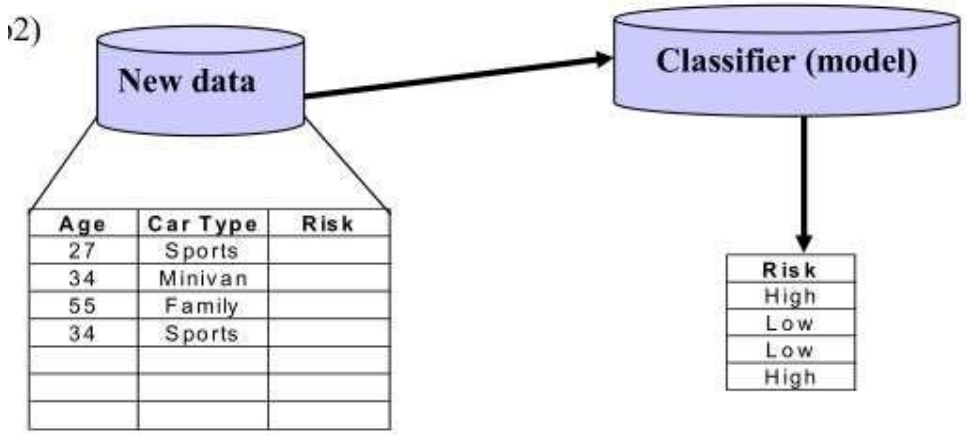
- Bước thứ hai (phân loại)

  Bước thứ hai của quá trình phân lớp dữ liệu là sử dụng mô hình đã học được ở bước trước để phân loại dữ liệu mới. Điều đầu tiên cần làm là ước tính độ

chính xác của mô hình phân lớp vừa được tạo ra. Holdout là một kỹ thuật đơn giản để ước tính độ chính xác này. Kỹ thuật này sử dụng một tập dữ liệu kiểm tra với các mẫu đã được gán nhãn lớp. Các mẫu này được chọn ngẫu nhiên và độc lập với các mẫu trong tập dữ liệu huấn luyện. Độ chính xác của mô hình trên tập dữ liệu kiểm tra đã được đưa ra dưới dạng tỷ lệ phần trăm của các mẫu trong tập dữ liệu kiểm tra được phân loại đúng bởi mô hình (so với thực tế). Nếu độ chính xác của mô hình được ước tính dựa trên tập dữ liệu huấn luyện, thì kết quả thu được có thể không đáng tin cậy bởi vì mô hình có thể quá vừa dữ liệu. Quá vừa dữ liệu là hiện tượng kết quả phân loại trùng khớp với dữ liệu thực tế do mô hình phân loại được học từ tập dữ liệu huấn luyện và có thể đã học được những đặc điểm riêng biệt của tập dữ liệu đó. Do đó, cần sử dụng một tập dữ liệu kiểm tra độc lập với tập dữ liệu huấn luyện để đánh giá độ chính xác của mô hình. Nếu độ chính xác của mô hình là chấp nhận được, thì mô hình được sử dụng để phân loại các dữ liệu tương lai hoặc các dữ liệu có giá trị thuộc tính phân loại chưa biết trước.



Hình 9: (b1)Ước lượng độ chính xác của mô hình



Hình 10: (b2)Phân lớp dữ liệu mới

2.2. Các vấn đề liên quan đến phân lớp dữ liệu

2.2.1. Chuẩn bị dữ liệu cho việc phân lớp

Việc tiền xử lý dữ liệu là vô cùng cần thiết và đóng vai trò quan trọng trong quá trình phân lớp, ảnh hưởng trực tiếp đến khả năng áp dụng được hay không của mô hình phân lớp. Quá trình tiền xử lý dữ liệu giúp tăng độ chính xác, hiệu quả và khả năng mở rộng của mô hình phân lớp.

Quá trình tiền xử lý dữ liệu là bước quan trọng để chuẩn bị dữ liệu cho quá trình phân tích. Nó bao gồm các công việc sau:

- Làm sạch dữ liệu: Làm sạch dữ liệu là quá trình loại bỏ thông tin không liên quan hoặc không chính xác trong dữ liệu. Việc làm sạch dữ liệu không chỉ đơn thuần là loại bỏ thông tin không cần thiết, mà còn bao gồm sửa chữa thông tin sai lệch và giảm số lượng bản sao trong tập dữ liệu.
- Phân tích tính cần thiết của dữ liệu: Trong tập dữ liệu, có rất nhiều thuộc tính không cần thiết hoặc không liên quan đến mục tiêu phân lớp. Việc phân tích tính cần thiết của dữ liệu giúp loại bỏ các thuộc tính không cần thiết, giúp quá trình phân tích nhanh hơn và hiệu quả hơn.
- Chuyển đổi dữ liệu: Việc khái quát hóa dữ liệu lên mức khái niệm cao hơn giúp cho các thuộc tính liên tục có thể được phân tích hiệu quả hơn.

2.2.2. So sánh các mô hình phân lớp

Trong từng ứng dụng cụ thể cần lựa chọn mô hình phân lớp phù hợp. Việc lựa chọn đó căn cứ vào sự so sánh các mô hình phân lớp với nhau, dựa trên các tiêu chuẩn sau:

- Độ chính xác dự đoán (predictive accuracy): Độ chính xác là khả năng của mô hình để dự đoán chính xác nhãn lớp của dữ liệu mới hay dữ liệu chưa biết.
- Tốc độ (speed): Tốc độ là những chi phí tính toán liên quan đến quá trình tạo ra và sử dụng mô hình.
- Sức mạnh (robustness): Sức mạnh là khả năng mô hình tạo ta những dự đoán đúng từ những dữ liệu noise hay dữ liệu với những giá trị thiếu.
- Khả năng mở rộng (scalability): Khả năng mở rộng là khả năng thực thi hiệu quả trên lượng lớn dữ liệu của mô hình đã học.
- Tính hiểu được (interpretability): Tính hiểu được là mức độ hiểu và hiểu rõ những kết quả sinh ra bởi mô hình đã học.
- Tính đơn giản (simplicity): Tính đơn giản liên quan đến kích thước của cây quyết định hay độ cô đọng của các luật.

2.3. Các phương thuật toán phân lớp dữ liệu

2.3.1. Thuật toán SVM

SVM (Support Vector Machine) là một thuật toán học máy giám sát được sử dụng cho các bài toán phân loại và hồi quy. SVM hoạt động bằng cách tìm ra đường phân chia tối ưu giữa các lớp dữ liệu khác nhau.

Trong phân loại, SVM tìm kiếm đường phân chia tối ưu giữa các lớp dữ liệu bằng cách xác định vector hỗ trợ, tức là các điểm dữ liệu nằm gần nhất với ranh giới của các lớp. Đường phân chia được xác định bằng cách tìm ra một siêu mặt phẳng (hyperplane) sao cho khoảng cách từ siêu mặt phẳng đến các hỗ trợ vector là lớn nhất. Các điểm dữ liệu mới có thể được phân loại bằng cách xác định vị trí của chúng đối với siêu mặt phẳng này.

Trong hồi quy, SVM tìm kiếm một đường phân chia tối ưu giữa các điểm dữ liệu bằng cách xác định hỗ trợ vector. Đường phân chia được xác định bằng cách tìm ra một siêu mặt phẳng (hyperplane) sao cho khoảng cách từ siêu mặt phẳng đến các điểm dữ liệu là lớn nhất.

Một số ưu điểm của SVM:

- Hiệu quả với các bộ dữ liệu lớn, đặc biệt là với số chiều lớn.
- Tính toán đơn giản và dễ triển khai.
- Phân loại chính xác trên các bộ dữ liệu phức tạp với nhiễu và đa dạng.

Một số nhược điểm của SVM là:

- Đòi hỏi phải xác định tham số đầu vào chính xác.
- SVM không thể giải quyết các bài toán phân loại có nhiều lớp dữ liệu.
- Không hiệu quả khi dữ liệu bị chồng chéo hoặc có sự chênh lệch lớn giữa các lớp.

SVM được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau với các ứng dụng đa dạng. Dưới đây là một số ví dụ về các ứng dụng của SVM:

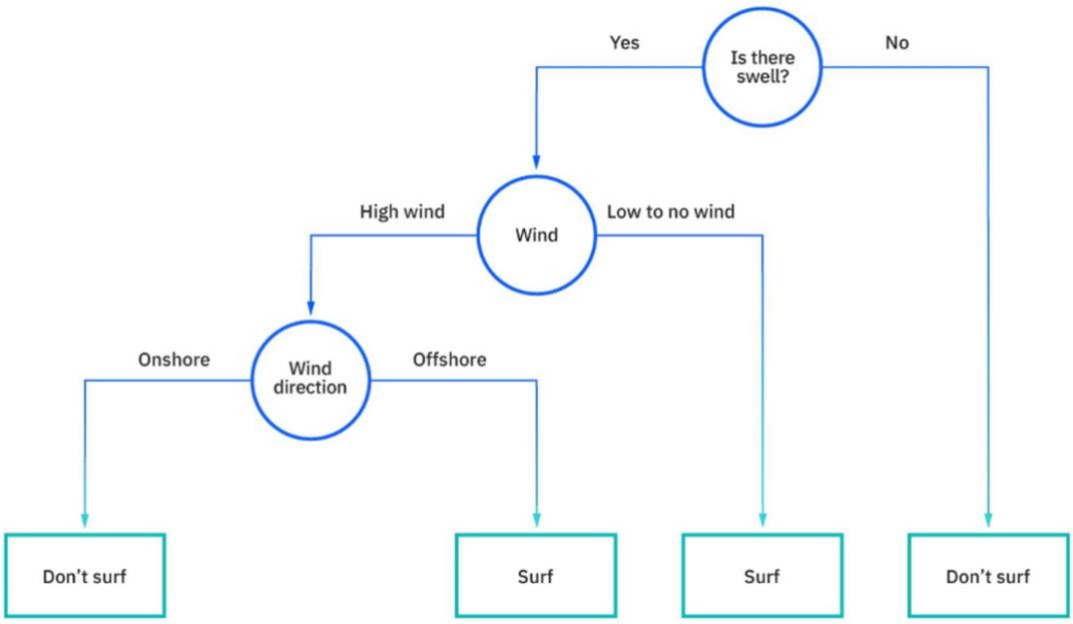
- Phân loại văn bản: SVM có thể được sử dụng để phân loại các tài liệu văn bản, ví dụ như email là thư rác hay không, phân loại tin tức vào các chủ đề khác nhau hoặc đánh giá cảm xúc của đoạn văn.
- Phát hiện giả mạo tiền: SVM có thể được sử dụng để phát hiện các tiền giả bằng cách phân loại các hình ảnh của tiền thật và giả.
- Phân loại ảnh: SVM có thể được sử dụng để phân loại các đối tượng trong ảnh, như phân loại các loài động vật hoặc phân loại các loại thực phẩm.
- Dự đoán giá cổ phiếu: SVM có thể được sử dụng để dự đoán giá cổ phiếu bằng cách phân tích các dữ liệu tài chính và kinh doanh.

- Nhận dạng tiếng nói: SVM có thể được sử dụng để phân loại các âm thanh thành các từ và câu, như là trong các hệ thống nhận dạng tiếng nói.
- Phân tích dữ liệu y tế: SVM có thể được sử dụng để phân tích dữ liệu y tế, ví dụ như phân loại các bệnh nhân vào các nhóm rủi ro khác nhau hoặc phân loại các tế bào ung thư.
- Phân loại mục tiêu tiếp thị: SVM có thể được sử dụng để phân loại khách hàng vào các nhóm khách hàng khác nhau để tăng cường chiến lược tiếp thị.

2.3.2. Thuật toán cây quyết định(Decision Tree)

Thuật toán Cây quyết định thuộc họ thuật toán học có giám sát. Không giống như các thuật toán học có giám sát khác, thuật toán cây quyết định cũng có thể được sử dụng để giải các bài toán hồi quy và phân loại .

Mục tiêu của việc sử dụng Cây quyết định là tạo ra một mô hình đào tạo có thể sử dụng để dự đoán loại hoặc giá trị của biến mục tiêu bằng cách học các quy tắc quyết định đơn giản được suy ra từ dữ liệu trước đó (dữ liệu đào tạo). Trong Cây quyết định, để dự đoán nhãn lớp cho bản ghi, bắt đầu từ gốc của cây. So sánh các giá trị của thuộc tính gốc với thuộc tính của bản ghi. Trên cơ sở so sánh, đi theo nhánh tương ứng với giá trị đó và nhảy sang nút tiếp theo.



Hình 11: Ví dụ về cây quyết định

Ý tưởng chính của thuật toán cây quyết định là tạo ra một cây phân loại các điểm dữ liệu dựa trên các thuộc tính của chúng. Các thuộc tính này được sử dụng để phân loại các điểm dữ liệu vào các nhóm khác nhau tùy thuộc vào giá trị của chúng. Thuật toán sẽ tự động xác định thuộc tính nào quan trọng hơn để chia các điểm dữ liệu thành các nhóm khác nhau.

Các loại cây quyết định

Các loại cây quyết định dựa trên loại biến mục tiêu mà chúng ta có. Nó có thể có hai loại:

- Cây quyết định biến phân loại: Cây quyết định có biến mục tiêu phân loại thì nó được gọi là cây quyết định biến phân loại.
- Cây quyết định biến liên tục: Cây quyết định có biến mục tiêu liên tục thì nó được gọi là Cây quyết định biến liên tục.

Thuật ngữ quan trọng liên quan đến cây quyết định

- Nút gốc: Nó đại diện cho toàn bộ dân số hoặc mẫu và điều này tiếp tục được chia thành hai hoặc nhiều bộ đồng nhất.
- Tách: Là quá trình chia một nút thành hai hoặc nhiều nút phụ.
- Nút quyết định: Khi một nút phụ tách thành các nút phụ khác, thì nó được gọi là nút quyết định.
- Lá/Nút đầu cuối: Các nút không phân chia được gọi là nút Lá hoặc nút đầu cuối.
- Cắt tỉa: Khi chúng ta loại bỏ các nút con của một nút quyết định, quá trình này được gọi là cắt tỉa. Bạn có thể nói quá trình phân tách ngược lại.
- Nhánh / Cây con: Một phần con của toàn bộ cây được gọi là nhánh hoặc cây con.
- Nút cha và nút con: Một nút được chia thành các nút con được gọi là nút cha của các nút con trong khi các nút con là nút con của nút cha.

Cây quyết định phân loại các ví dụ bằng cách sắp xếp chúng xuống cây từ gốc đến một số nút lá/đầu cuối, với nút lá/đầu cuối cung cấp phân loại của ví dụ. Mỗi nút trong cây hoạt động như một ca kiểm thử cho một số thuộc tính và mỗi cạnh đi xuống từ nút tương ứng với các câu trả lời có thể có cho ca kiểm thử. Quá trình này có tính chất đệ quy và được lặp lại cho mọi cây con bắt nguồn từ nút mới.

Cây quyết định tuân theo biểu diễn Tổng sản phẩm (SOP). Tổng của sản phẩm (SOP) còn được gọi là Dạng chuẩn riêng biệt. Đối với một lớp, mọi nhánh từ gốc của cây đến nút lá có cùng một lớp là tập hợp (tích) của các giá trị, các nhánh khác nhau kết thúc trong lớp đó tạo thành một phép chia (tổng). Thách thức chính trong việc triển khai cây quyết định là xác định thuộc tính nào chúng ta cần xem xét làm nút gốc và từng cấp độ. Xử lý điều này là để biết như các thuộc tính lựa chọn. Chúng tôi có các biện pháp lựa chọn thuộc tính khác nhau để xác định thuộc tính có thể được coi là nốt gốc ở mỗi cấp độ.

Cây quyết định hoạt động như thế nào?

Quyết định phân chia chiến lược ảnh hưởng rất nhiều đến độ chính xác của cây. Các tiêu chí quyết định là khác nhau đối với cây phân loại và cây hồi quy. Cây quyết định sử dụng nhiều thuật toán để quyết định chia một nút thành hai hoặc nhiều nút phụ.

Việc tạo các nút phụ làm tăng tính đồng nhất của các nút phụ kết quả. Nói cách khác, chúng ta có thể nói rằng độ tinh khiết của nút tăng lên đối với biến mục tiêu. Cây quyết định phân tách các nút trên tất cả các biến có sẵn và sau đó chọn phân tách dẫn đến hầu hết các nút con đồng nhất.

Việc lựa chọn thuật toán cũng dựa trên loại biến mục tiêu. Chúng ta hãy xem xét một số thuật toán được sử dụng trong Cây quyết định:

- Extension of D3 (ID3): Ross Quinlan được ghi nhận trong quá trình phát triển ID3. Thuật toán này tận dụng entropy và mức tăng thông tin làm số liệu để đánh giá các phân tách ứng cử viên.
- Successor of ID3 (C4.5): Thuật toán này được coi là phiên bản sau của ID3, cũng được phát triển bởi Quinlan. Nó có thể sử dụng thông tin đạt được hoặc tỷ lệ đạt được để đánh giá các điểm phân chia trong cây quyết định.
- Classification And Regression Tree (CART): là chữ viết tắt của "cây phân loại và hồi quy" và được giới thiệu bởi Leo Breiman. Thuật toán này thường sử dụng Gini để xác định thuộc tính lý tưởng để phân tách. Gini đo tần suất một thuộc tính được chọn ngẫu nhiên bị phân loại sai. Khi đánh giá bằng cách sử Gini, giá trị thấp hơn sẽ lý tưởng hơn.
- Chi-square automatic interaction detection Performs multi-level splits when computing classification trees (CHAID)
- Multivariate adaptive regression splines (MARS)

Chỉ số Gini

Dùng trong thuật toán CART (Classification and Regression Trees). Nó dựa vào việc bình phương các xác suất thành viên cho mỗi thể loại đích trong nút. Giá trị của nó tiến đến cực tiểu (bằng 0) khi mọi trường hợp trong nút rơi vào một thể loại đích duy nhất.

Có thể hiểu chỉ số Gini là một hàm chi phí dùng để đánh giá sự phân chia trong tập dữ liệu. Nó được tính bằng cách trừ đi tổng các xác suất bình phương của mỗi lớp từ một. Nó ưu tiên các phân vùng lớn hơn và dễ thực hiện trong khi thông tin thu được ưu tiên các phân vùng nhỏ hơn với các giá trị riêng biệt. Chỉ số Gini hoạt động với biến mục tiêu phân loại "Thành công" hoặc "Thất bại". Nó chỉ thực hiện phân chia nhị phân.

Chỉ số Gini là xác suất phân loại sai điểm dữ liệu ngẫu nhiên trong tập dữ liệu nếu nó được gắn nhãn dựa trên phân phối lớp của tập dữ liệu. Tương tự như entropy, nếu tập S là thuần túy—tức là thuộc về một lớp) thì chỉ số của nó bằng không. Điều này được biểu thị bằng công thức sau:

$$\text{Gini} = 1 - \sum_{i=1}^{j} P(i)^2$$

Các bước để tính chỉ số Gini cho một lần chia tách:

- Tính Gini cho các nút phụ, sử dụng công thức trên cho thành công(p) và thất bại(q) (p^2+q^2).
- Tính toán chỉ số Gini cho quá trình phân tách bằng cách sử dụng điểm Gini có trọng số của từng nút trong quá trình phân tách đó.

Entropy

Dùng trong các thuật toán sinh cây ID3, C4.5 và C5.0. Số đo này dựa trên khái niệm entropy trong lý thuyết thông tin (information theory).

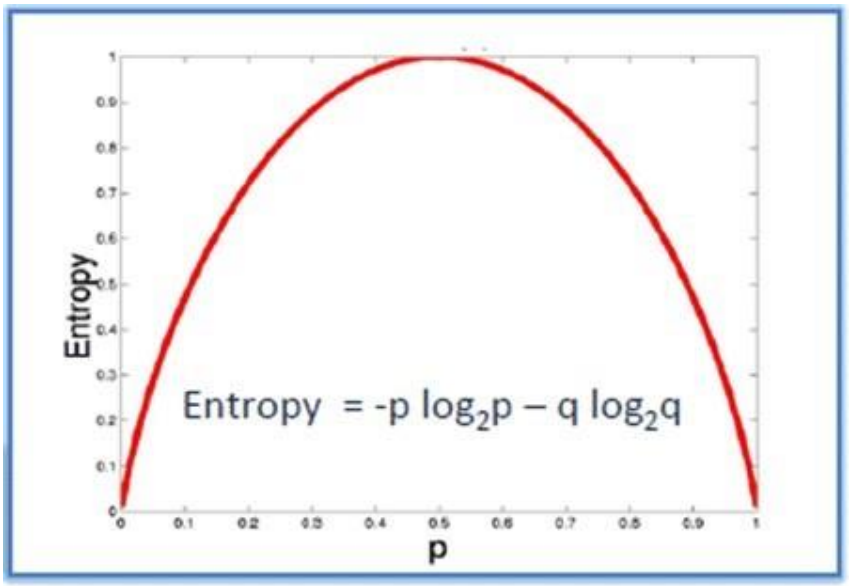
Entropy là thuật ngữ thuộc Nhiệt động lực học, nó đo lường mức độ hỗn loạn hay sự ngẫu nhiên trong một tập dữ liệu. Năm 1948, Claude Shannon đã mở rộng khái niệm entropy sang lĩnh vực thống kê, với công thức tính toán như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2, \ldots, x_n$.

Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x = x_i)$.

Ký hiệu phân phối này là $p = (p_1, p_2, \ldots, p_n)$. Entropy của phân phối này được định nghĩa là:

$$H_{(p)} = -\sum_{n=1}^{n} p_i \log(p_i)$$



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Hình 12: Hàm Entropy

Hình trên biểu diễn sự thay đổi của hàm entropy. Ta có thể thấy rằng, entropy đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

- P tinh khiết: $p_i = 0$ hoặc $p_i = 1$
- P vẩn đục: $p_i = 0.5$, khi đó hàm Entropy đạt đỉnh cao nhất

Information Gain trong Cây quyết định (Decision Tree)

Information Gain được sử dụng trong các thuật toán sinh cây quyết định và dựa trên việc giảm Entropy của tập dữ liệu sau khi được chia trên một thuộc tính. Để xây dựng một cây quyết định, chúng ta cần tìm thuộc tính có Information Gain cao nhất.

Để xác định các nút trong cây quyết định, ta thực hiện tính Information Gain tại mỗi nút theo các bước sau:

- Bước 1: Tính toán Entropy của biến mục tiêu S với N phần tử và số phần tử thuộc lớp c là N_c.

$$H_{(S)} = -\sum^{c}_{c=1} (N_c / N) \log(N_c / N)$$

- Bước 2: Tính hàm Entropy tại mỗi thuộc tính x. Với thuộc tính x, ta chia tập dữ liệu S thành K child node S_1, S_2, ..., S_K với số điểm dữ liệu trong mỗi child node lần lượt là m_1, m_2, ..., m_K. Hàm Entropy của thuộc tính x được tính bằng công thức:

$$H_{(x, S)} = \sum^{K}_{k=1} (m_k / N) * H_(S_k)$$

- Bước 3: Tính chỉ số Gain Information:

$$G_{(x, S)} = H_{(S)} - H_{(x,S)}$$

Tiêu chuẩn dừng

Trong các thuật toán Decision tree, phương pháp chia trên sẽ liên tục chia các nút cho đến khi chúng trở thành tinh khiết. Điều này dẫn đến việc cây quyết định có thể trở nên rất phức tạp với nhiều nút lá chỉ có một vài điểm dữ liệu. Mặc dù đưa ra dự đoán chính xác trên tập huấn luyện (với giả định rằng không có hai đầu vào nào cho kết quả khác nhau), nhưng cây có thể dẫn đến hiện tượng quá khớp (overfitting) trong việc dự đoán trên tập kiểm tra.

Để tránh tình trạng overfitting trong các thuật toán Decision Tree, ta có thể ngừng việc phân chia cây dựa trên một số phương pháp sau:

- Nếu node đó có hệ số entropy bằng 0, tức là tất cả các điểm dữ liệu trong node đó đều thuộc vào một lớp duy nhất.

- Nếu node đó có số lượng điểm dữ liệu nhỏ hơn một ngưỡng cố định, trong trường hợp này, ta chấp nhận một số điểm dữ liệu bị phân lớp sai để tránh overfitting. Lớp cho leaf node này có thể được xác định dựa trên lớp chiếm đa số trong node.
- Nếu khoảng cách từ node đó đến root node đạt đến một giá trị cố định. Việc giới hạn độ sâu của cây giảm độ phức tạp của cây và hỗ trợ tránh overfitting.
- Nếu tổng số leaf node vượt quá một ngưỡng cố định.
- Nếu phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng cố định).

Thuật toán cây quyết định có nhiều ưu điểm như sau:

- Dễ hiểu: Các quy tắc của cây quyết định dễ diễn giải và biểu diễn trực quan, giúp người dùng dễ sử dụng hơn. Bản chất phân cấp của cây quyết định cũng giúp xác định thuộc tính quan trọng nhất dễ dàng hơn, điều này không phải lúc nào cũng đơn giản với các thuật toán khác như mạng nơ-ron.
- Linh hoạt với dữ liệu: Cây quyết định có khả năng xử lý nhiều loại dữ liệu khác nhau, bao gồm cả dữ liệu rời rạc và liên tục. Các giá trị liên tục có thể được chuyển đổi thành các giá trị phân loại thông qua sử dụng ngưỡng, và cây quyết định cũng có thể xử lý các giá trị bị thiếu. Điều này khác biệt với các bộ phân loại khác, chẳng hạn như Naïve Bayes.
- Linh hoạt với các nhiệm vụ: Cây quyết định có thể được sử dụng cho cả nhiệm vụ phân loại và hồi quy, làm cho nó linh hoạt hơn một số thuật toán khác. Nó cũng không nhạy cảm với các mối quan hệ cơ bản giữa các thuộc tính; điều này có nghĩa là nếu hai biến có tương quan cao, thuật toán sẽ chỉ chọn một trong các tính năng để phân tách.

Tuy nhiên, cây quyết định cũng có một số nhược điểm:

- Dễ bị quá khớp: Các cây quyết định phức tạp có xu hướng bị quá khớp và không tổng quát hóa tốt cho dữ liệu mới. Để tránh tình trạng này, ta có thể sử dụng các quá trình cắt tỉa trước hoặc sau khi tạo cây. Cắt tỉa trước tạm dừng sự phát triển của cây khi không đủ dữ liệu, trong khi cắt tỉa sau loại bỏ các cây con có dữ liệu không đầy đủ sau khi tạo cây.
- Công cụ ước tính phương sai cao: Các biến thể nhỏ trong dữ liệu có thể tạo ra một cây quyết định rất khác. Đóng gói, hoặc tính trung bình của các ước tính, có thể là một phương pháp giảm phương sai của cây quyết định. Tuy nhiên, phương pháp này có hạn chế vì nó có thể dẫn đến các yếu tố dự đoán tương quan cao.
- Tốn kém hơn: Việc đào tạo cây quyết định có thể tốn kém hơn so với các thuật toán khác, do cây quyết định sử dụng phương pháp tìm kiếm tham lam trong quá trình xây dựng.
- Không được hỗ trợ đầy đủ trong scikit-learn: Scikit-learn là một thư viện phổ biến về học máy dựa trên Python. Mặc dù thư viện này có mô-đun Cây quyết định, nhưng việc triển khai hiện tại không hỗ trợ các biến phân loại.

Thuật toán cây quyết định (Decision Tree) được sử dụng rộng rãi trong các ứng dụng của Học máy và Thị giác máy tính, bao gồm:

- Phân loại: Thuật toán cây quyết định có thể được sử dụng để phân loại các đối tượng vào các nhóm khác nhau, như phân loại các loài hoa, xác định loại xe hơi dựa trên các đặc tính kỹ thuật, hoặc phân loại email là spam hay không.
- Dự đoán: Thuật toán cây quyết định có thể được sử dụng để dự đoán giá cổ phiếu, dự đoán giá nhà đất hoặc dự đoán thu nhập của một người dựa trên các thuộc tính như tuổi, trình độ học vấn và nghề nghiệp.
- Tạo ra luật: Thuật toán cây quyết định có thể được sử dụng để tạo ra các luật để mô tả mối quan hệ giữa các thuộc tính và các kết quả. Các luật này có thể được sử dụng để giải thích các quyết định được đưa ra bởi hệ thống.
- Xử lý dữ liệu bị thiếu: Thuật toán cây quyết định có thể được sử dụng để xử lý các dữ liệu bị thiếu, bằng cách thay thế các giá trị bị thiếu bằng giá trị trung bình hoặc giá trị phổ biến nhất của các thuộc tính.
- Phát hiện bất thường: Thuật toán cây quyết định có thể được sử dụng để phát hiện các điểm dữ liệu bất thường, bằng cách xây dựng các cây quyết định khác nhau cho các tập con của dữ liệu và so sánh các cây quyết định này với nhau.

2.3.3. Logistic Regression (Hồi quy Logistic)

Thuật toán Logistic Regression là một thuật toán học máy thuộc loại phân loại nhị phân (binary classification), tức là chỉ có hai nhóm được phân loại. Thuật toán này dựa trên việc ước lượng xác suất của một mẫu thuộc về một nhóm.

Cơ chế hoạt động của thuật toán Logistic Regression là tìm một đường cong logistic (sigmoid function) để phân loại các mẫu vào hai nhóm, thông qua việc tối ưu hóa hàm mất mát (loss function) dựa trên việc so sánh giữa giá trị dự đoán và giá trị thực tế.

Có ba loại mô hình hồi quy logistic, được xác định dựa trên phản ứng phân loại:

- Hồi quy logistic nhị phân: Theo cách tiếp cận này, phản hồi hoặc biến phụ thuộc về bản chất là phân đôi—tức là nó chỉ có hai kết quả có thể xảy ra (ví dụ: 0 hoặc 1). Một số ví dụ phổ biến về việc sử dụng nó bao gồm dự đoán xem một e-mail có phải là thư rác hay không phải thư rác hoặc khối u ác tính hay không ác tính. Trong hồi quy logistic, đây là cách tiếp cận được sử dụng phổ biến nhất và nói chung, nó là một trong những cách phân loại phổ biến nhất để phân loại nhị phân.
- Hồi quy logistic đa thức: Trong loại mô hình hồi quy logistic này, biến phụ thuộc có ba hoặc nhiều kết quả có thể xảy ra; tuy nhiên, các giá trị này không có thứ tự cụ thể. Ví dụ: các hãng phim muốn dự đoán thể loại phim mà khán giả có khả năng xem để tiếp thị phim hiệu quả hơn. Mô hình hồi quy logistic đa thức có thể giúp hãng phim xác định mức độ ảnh hưởng của tuổi tác, giới tính và tình trạng hẹn hò của một người đối với loại phim họ thích. Sau đó, hãng phim có thể định

hướng chiến dịch quảng cáo của một bộ phim cụ thể tới một nhóm người có khả năng sẽ đi xem bộ phim đó.

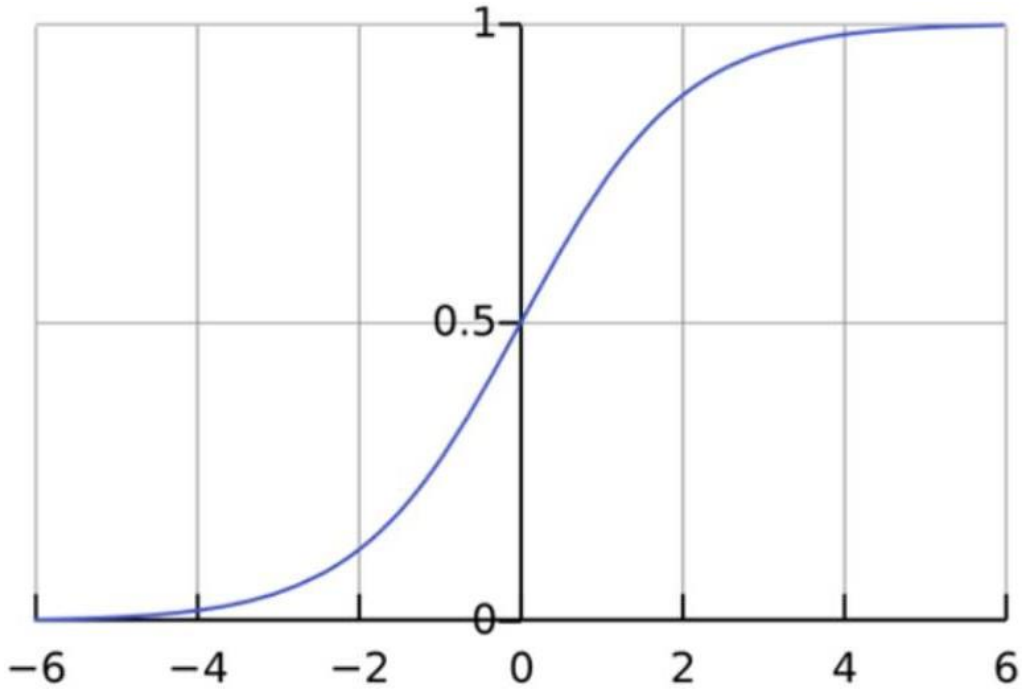
- Hồi quy logistic thông thường: Loại mô hình hồi quy logistic này được tận dụng khi biến phản hồi có ba hoặc nhiều hơn kết quả có thể xảy ra, nhưng trong trường hợp này, các giá trị này có một thứ tự xác định. Ví dụ về các câu trả lời theo thứ tự bao gồm thang điểm từ A đến F hoặc thang đánh giá từ 1 đến 5.

Hồi quy Logistic hoạt động như thế nào:

Để hiểu về cách hoạt động của mô hình hồi quy logistic, ta cần hiểu các khái niệm sau:

- Phương trình: Đó là mối quan hệ giữa hai biến x và y trong toán học. Chúng ta có thể sử dụng phương trình hoặc hàm để vẽ đồ thị theo trục x và trục y bằng cách nhập các giá trị khác nhau của x và y. Ví dụ, nếu ta vẽ đồ thị cho hàm y = 2*x, ta sẽ có một đường thẳng tuyến tính.
- Biến: Trong thống kê, biến là các yếu tố dữ liệu hoặc thuộc tính có giá trị khác nhau. Một số biến nhất định được gọi là biến độc lập hoặc biến giải thích. Những thuộc tính này là nguyên nhân của một kết quả. Các biến khác là biến phụ thuộc hoặc biến đáp ứng; giá trị của chúng phụ thuộc vào các biến độc lập. Hồi quy logistic khám phá cách các biến độc lập ảnh hưởng đến một biến phụ thuộc bằng cách xem xét các giá trị dữ liệu lịch sử của cả hai biến.
- Hàm hồi quy logistic: Đây là một mô hình thống kê sử dụng hàm logistic, hay hàm logit trong toán học, làm phương trình giữa hai biến x và y. Hàm logit ánh xạ y thành hàm sigmoid của x.

$$f(x) = \frac{1}{1+ e^{-x}}$$



Hình 13: Đồ thị phương trình hồi quy logistic

Hồi quy logistic là phương pháp được sử dụng để ước tính giá trị của biến phụ thuộc, và hàm logit được sử dụng để chuyển đổi giá trị của biến phụ thuộc thành các giá trị nằm trong khoảng từ 0 đến 1, bất kể giá trị của biến độc lập là gì. Ngoài ra, phương pháp này cũng lập mô hình phương trình giữa nhiều biến độc lập và một biến phụ thuộc.

Phân tích hồi quy logistic với nhiều biến độc lập: Trong nhiều trường hợp, nhiều biến giải thích ảnh hưởng đến giá trị của biến phụ thuộc. Để lập mô hình các tập dữ liệu đầu vào như vậy, công thức hồi quy logistic phải giả định mối quan hệ tuyến tính giữa các biến độc lập khác nhau. Có thể sửa đổi hàm sigmoid và tính toán biến đầu ra cuối cùng như sau:

$$y = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_n x_n)$$

Ký hiệu β đại diện cho hệ số hồi quy. Mô hình logit có thể đảo ngược tính toán các giá trị hệ số này khi bạn cho nó một tập dữ liệu thực nghiệm đủ lớn có các giá trị đã xác định của cả hai biến phụ thuộc và biến độc lập.

- Log của tỷ số odds: Mô hình logit cũng có thể xác định tỷ số thành công trên thất bại hay log của tỷ số odds. Về mặt toán học, tỷ số odds về mặt xác suất là $\frac{p}{1-p}$ và log của tỷ số odds là $\log \frac{p}{1-p}$. Biểu diễn hàm logistic bằng log của tỷ số odds như hình dưới đây:

$$\text{Logit Function} = \log \frac{p}{1-p}$$

Một số đặc điểm của thuật toán Logistic Regression bao gồm:

- Dễ hiểu và áp dụng: Thuật toán Logistic Regression rất đơn giản và dễ hiểu, có thể được áp dụng cho các bài toán phân loại đơn giản.
- Hiệu quả trong một số bài toán phân loại nhị phân: Logistic Regression là một trong những thuật toán phân loại nhị phân hiệu quả nhất và thường được sử dụng trong các bài toán phân loại đơn giản.
- Yêu cầu dữ liệu phân bố tuyến tính: Logistic Regression yêu cầu dữ liệu phân bố tuyến tính và không hoạt động tốt trong trường hợp dữ liệu không phân bố tuyến tính.
- Dễ bị overfitting: Thuật toán Logistic Regression có thể dễ bị overfitting khi áp dụng cho các bài toán phân loại phức tạp.

Các ưu điểm của thuật toán Logistic Regression:

- Dễ dàng hiểu và triển khai: Thuật toán Logistic Regression đơn giản và dễ dàng hiểu, do đó nó có thể triển khai một cách nhanh chóng và đơn giản.
- Tính toán nhanh: Do không có quá nhiều tham số, Logistic Regression thường được tính toán nhanh hơn so với các thuật toán khác.
- Độ chính xác cao với dữ liệu tuyến tính: Logistic Regression là thuật toán phân loại tuyến tính và hoạt động tốt với các bộ dữ liệu tuyến tính.

Các nhược điểm của thuật toán Logistic Regression:

- Dễ bị ảnh hưởng bởi nhiễu: Nếu có nhiều nhiễu hoặc outlier trong bộ dữ liệu, Logistic Regression có thể cho kết quả không chính xác.
- Khó xử lý các vấn đề phức tạp: Logistic Regression không thể xử lý các bài toán phức tạp, ví dụ như việc phân loại dữ liệu không tuyến tính.
- Yêu cầu dữ liệu phải chuẩn hóa: Để tăng độ chính xác, dữ liệu cần phải được chuẩn hóa, tuy nhiên điều này có thể làm giảm hiệu suất của thuật toán khi xử lý các bộ dữ liệu lớn.

Các ứng dụng của thuật toán Logistic Regression bao gồm:

- Phân loại email: Logistic Regression có thể được sử dụng để phân loại email vào các thư mục khác nhau, chẳng hạn như thư rác và thư quan trọng.
- Dự đoán khả năng mua hàng: Logistic Regression có thể được sử dụng để dự đoán khả năng mua hàng của khách hàng dựa trên các thông tin về khách hàng.
- Phân loại chủ đề văn bản: Logistic Regression có thể được sử dụng để phân loại các chủ đề văn bản khác nhau, chẳng hạn như phân loại tin tức vào các chủ đề khác nhau.
- Dự đoán khả năng trả nợ: Logistic Regression có thể được sử dụng để dự đoán khả năng trả nợ của khách hàng dựa trên các thông tin tài chính của họ.

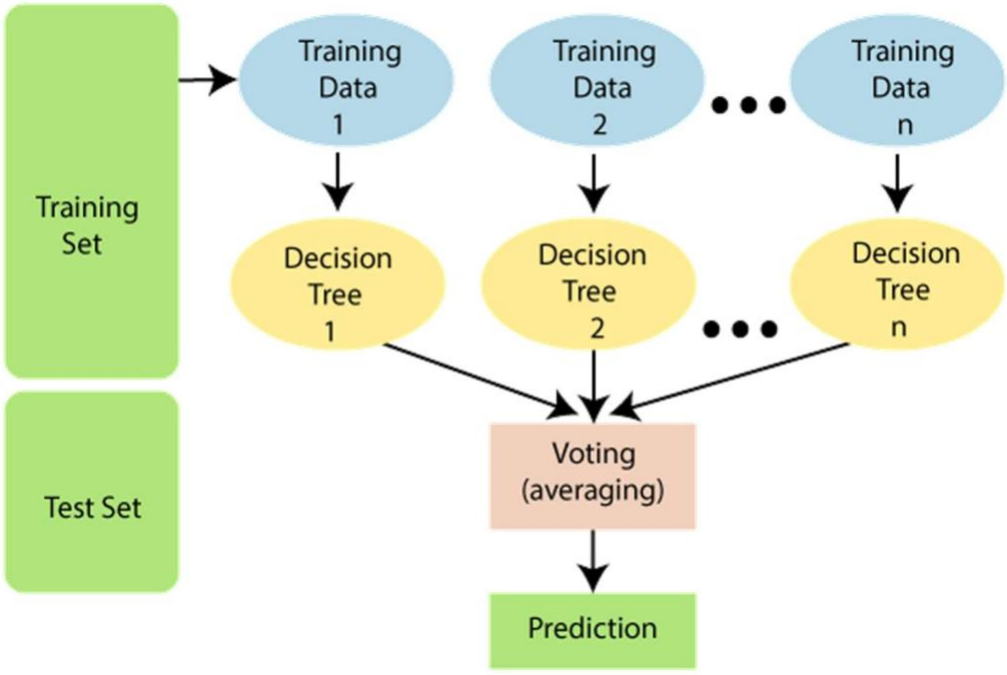
– Phân loại hình ảnh: Logistic Regression có thể được sử dụng để phân loại hình.

1.2.4. Thuật toán rừng cây ngẫu nhiên(Ramdom forest)

Thuật toán rừng cây ngẫu nhiên (Random Forest) là một phương pháp học máy dựa trên việc kết hợp nhiều cây quyết định (decision tree) để tạo ra một mô hình dự đoán chính xác hơn và tránh tình trạng overfitting.

Thuật toán Random Forest hoạt động bằng cách xây dựng nhiều cây quyết định độc lập nhau trên các tập con của tập dữ liệu huấn luyện, với các tập con được chọn ngẫu nhiên và không trùng lặp. Mỗi cây quyết định được xây dựng bằng cách chọn một số lượng nhỏ các thuộc tính ngẫu nhiên từ tất cả các thuộc tính của tập dữ liệu huấn luyện, và sử dụng chúng để xây dựng cây quyết định. Khi dự đoán, kết quả của các cây quyết định được kết hợp lại với nhau để đưa ra kết quả dự đoán cuối cùng.

Thuật toán rừng cây ngẫu nhiên là một phần mở rộng của phương pháp đóng bao vì nó sử dụng cả tính ngẫu nhiên đóng bao và tính năng đặc trưng để tạo ra một rừng cây quyết định không tương quan. Tính ngẫu nhiên của tính năng, còn được gọi là đóng gói tính năng hoặc " phương pháp không gian con ngẫu, tạo ra một tập hợp con ngẫu nhiên các tính năng, đảm bảo mức độ tương quan thấp giữa các cây quyết định. Đây là điểm khác biệt chính giữa cây quyết định và rừng ngẫu nhiên. Trong khi các cây quyết định xem xét tất cả các phân tách tính năng có thể có, thì các khu rừng ngẫu nhiên chỉ chọn một tập hợp con của các tính năng đó.



Hình 14: Sơ đồ hoạt động của thuật toán Random Forest

Cách hoạt động của thuật toán Random Forest:

Các thuật toán rừng ngẫu nhiên có ba siêu tham số chính cần được thiết lập trước khi huấn luyện. Chúng bao gồm kích thước nút, số lượng cây và số lượng tính năng được lấy mẫu. Từ đó, bộ phân loại rừng ngẫu nhiên có thể được sử dụng để giải quyết các vấn đề hồi quy hoặc phân loại.

Thuật toán rừng ngẫu nhiên được tạo thành từ một tập hợp các cây quyết định và mỗi cây trong tập hợp bao gồm một mẫu dữ liệu được lấy từ một tập huấn luyện có thay thế, được gọi là mẫu bootstrap. Trong số mẫu đào tạo đó, một phần ba trong số đó được dành làm dữ liệu thử nghiệm, được gọi là mẫu xuất xưởng (oob. Một ví dụ khác về tính ngẫu nhiên sau đó được đưa vào thông qua tính năng đóng gói, bổ sung thêm tính đa dạng cho tập dữ liệu và giảm mối tương quan giữa các cây quyết định. Tùy thuộc vào loại vấn đề, việc xác định dự đoán sẽ khác nhau. Đối với nhiệm vụ hồi quy, các cây quyết định riêng lẻ sẽ được tính trung bình và đối với nhiệm vụ phân loại, đa số phiếu bầu - tức là biến phân loại thường xuyên nhất - sẽ mang lại lớp dự đoán. Cuối cùng, mẫu oob sau đó được sử dụng để xác thực chéo.

Ưu điểm của thuật toán Random Forest:

- Độ chính xác cao: Random Forest là một trong những thuật toán học máy chính xác và hiệu quả nhất, đặc biệt là đối với các tập dữ liệu lớn.
- Điều khiển overfitting: Do sử dụng nhiều cây quyết định, Random Forest có khả năng tránh tình trạng overfitting, tức là mô hình không chỉ tìm hiểu các đặc điểm của tập dữ liệu huấn luyện mà còn có khả năng dự đoán tốt với các dữ liệu mới.
- Xử lý tập dữ liệu lớn: Thuật toán Random Forest có khả năng xử lý các tập dữ liệu lớn, có số lượng đặc trưng lớn và khó tính toán.

Tuy nhiên, Random Forest cũng có một số nhược điểm:

- Tốn thời gian huấn luyện: Do phải xây dựng nhiều cây quyết định, Random Forest có thể tốn nhiều thời gian để huấn luyện.
- Không thể giải thích được quá trình quyết định: Random Forest không cung cấp một giải thích rõ ràng cho quá trình quyết định, do đó khó để hiểu được lý do tại sao một dự đoán được đưa ra.

Sự khác biệt giữa Cây quyết định và Rừng ngẫu nhiên

Cây quyết định	Rừng cây ngẫu nhiên
Cây quyết định thường gặp phải vấn đề trang bị quá mức nếu nó được phép phát triển mà không có bất kỳ sự kiểm soát nào.	Các khu rừng ngẫu nhiên được tạo từ các tập hợp con dữ liệu và kết quả cuối cùng dựa trên xếp hạng trung bình hoặc đa số; do đó vấn đề overfitting được quan tâm.

Tính toán một cây quyết định đơn lẻ nhanh hơn.	Nó tương đối chậm hơn.
Khi một tập dữ liệu với các tính năng được cây quyết định lấy làm đầu vào, nó sẽ hình thành một số quy tắc để đưa ra dự đoán.	Rừng ngẫu nhiên chọn ngẫu nhiên các quan sát, xây dựng cây quyết định và lấy kết quả trung bình. Nó không sử dụng bất kỳ bộ công thức nào.

Bảng 1: So sánh cây quyết định và rừng cây ngẫu nhiên

Một số ứng dụng của thuật toán rừng cây ngẫu nhiên:

- Phân loại hình ảnh: Random Forest được sử dụng để phân loại hình ảnh trong nhiều lĩnh vực như y học, khoa học nông nghiệp và công nghiệp.
- Dự đoán giá trị tài sản: Random Forest có thể được sử dụng để dự đoán giá trị tài sản, giúp các nhà đầu tư và công ty quản lý tài sản đưa ra quyết định đầu tư và quản lý tài sản hiệu quả.
- Phát hiện gian lận tín dụng: Random Forest được sử dụng để phát hiện các hành vi gian lận trong việc sử dụng thẻ tín dụng hoặc các hoạt động tài chính khác.
- Dự đoán khả năng sinh tồn của bệnh nhân: Random Forest được sử dụng trong y học để dự đoán khả năng sinh tồn của bệnh nhân và giúp bác sĩ đưa ra quyết định điều trị.
- Xác định chất lượng sản phẩm: Random Forest được sử dụng trong ngành sản xuất để xác định chất lượng sản phẩm và giúp tối ưu hóa quy trình sản xuất.
- Phân loại email: Random Forest được sử dụng để phân loại email, giúp người dùng quản lý email một cách hiệu quả hơn.
- Phân tích ngôn ngữ tự nhiên: Random Forest được sử dụng trong xử lý ngôn ngữ tự nhiên để phân tích các bài báo, đánh giá sản phẩm, đánh giá dịch vụ khách hàng và hỗ trợ chatbot.

1.2.5. Thuật toán K-Nearest Neighbors

Thuật toán k-láng giềng gần nhất, còn được gọi là KNN hoặc k-NN, là một thuật toán phân loại học có giám sát, phi tham số, sử dụng khoảng cách gần để phân loại hoặc dự đoán về việc nhóm một điểm dữ liệu riêng lẻ. Mặc dù nó có thể được sử dụng cho các vấn đề hồi quy hoặc phân loại, nhưng nó thường được sử dụng như một thuật toán phân loại, dựa trên giả định rằng các điểm tương tự có thể được tìm thấy gần nhau.

Cụ thể, để phân loại một điểm dữ liệu mới, thuật toán KNN sẽ tìm ra k điểm dữ liệu gần nhất với điểm đó trong không gian đặc trưng (feature space) bằng cách tính khoảng cách Euclide giữa chúng. Sau đó, thuật toán sẽ dự đoán nhãn của điểm dữ liệu mới bằng cách lấy nhãn xuất hiện nhiều nhất trong k điểm láng giềng này.

Với KNN, được sử dụng trong bài toán phân loại (Classification). Với thuật toán này, nhãn của một điểm dữ liệu mới (hoặc kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập huấn luyện (training set). Các nhãn này có thể được quyết định bằng cách bầu chọn theo số phiếu giữa các điểm gần nhất, hoặc bằng cách đánh trọng số khác nhau cho mỗi điểm gần nhất và suy ra nhãn từ đó.

Hình 15: Sơ đồ thuật toán KNN

Trong bài toán Regression, KNN tìm đầu ra của một điểm dữ liệu mới bằng cách sử dụng thông tin từ K điểm dữ liệu gần nhất trong tập huấn luyện. Nếu K=1, đầu ra của điểm dữ liệu mới sẽ bằng đầu ra của điểm dữ liệu gần nhất đó. Nếu K>1, đầu ra của điểm dữ liệu mới có thể được tính toán bằng trung bình có trọng số của đầu ra của những điểm gần nhất hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó. KNN không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này có thể là nhiễu.

Để xác định điểm dữ liệu nào gần nhất với một điểm truy vấn nhất định, cần phải tính toán khoảng cách giữa điểm truy vấn và các điểm dữ liệu khác. Các số liệu khoảng cách này giúp hình thành các ranh giới quyết định, phân vùng các điểm truy vấn thành các vùng khác nhau. Thường sẽ thấy các ranh giới quyết định được hiển thị bằng sơ đồ Voronoi. Đây là 4 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:

- Khoảng cách Euclide: Đây là thước đo khoảng cách được sử dụng phổ biến nhất và nó được giới hạn ở các vectơ có giá trị thực. Nó đo một đường thẳng giữa điểm truy vấn và điểm khác được đo.

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

– Khoảng cách Manhattan : Đây cũng là một thước đo khoảng cách phổ biến khác, đo giá trị tuyệt đối giữa hai điểm.

$$d(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

– Khoảng cách Minkowski : Phép đo khoảng cách này là dạng tổng quát của phép đo khoảng cách Euclidean và Manhattan. Tham số, p, trong công thức bên dưới, cho phép tạo các số liệu khoảng cách khác. Khoảng cách Euclide được biểu thị bằng công thức này khi p bằng hai và khoảng cách Manhattan được biểu thị bằng p bằng một.

$$\text{Minkowski: } \left(\sum_{i=1}^{k} (|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

– Khoảng cách Hamming: Kỹ thuật này thường được sử dụng với các vectơ Boolean hoặc chuỗi, xác định các điểm mà các vectơ không khớp. Do đó, nó còn được gọi là chỉ số trùng lặp.

$$\text{Hamming} = D_H = \left(\sum_{i=1}^{k} |x_i - y_i|\right)$$

Ưu điểm của thuật toán KNN bao gồm:

– Độ chính xác cao: KNN thường cho kết quả phân loại tốt với dữ liệu đầu vào là số lượng lớn.
– Dễ dàng triển khai: Thuật toán KNN rất đơn giản và dễ hiểu, và có thể được triển khai nhanh chóng với các thư viện học máy như scikit-learn của Python.

Thuật toán KNN cũng có một số nhược điểm như:

– Chi phí tính toán cao: Việc tính toán khoảng cách Euclide giữa một điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện là tốn kém và cần phải được thực hiện nhiều lần.
– Nhạy cảm với dữ liệu nhiễu: KNN có thể dễ bị ảnh hưởng bởi các điểm dữ liệu nhiễu hoặc các giá trị ngoại lai, do đó có thể dẫn đến kết quả phân loại không chính xác.

Các ứng dụng của thuật toán KNN là:

– Phân loại hình ảnh: KNN có thể được sử dụng để phân loại hình ảnh với độ chính xác cao, chẳng hạn như phân loại các loại hoa dựa trên hình ảnh của chúng.
– Dự đoán giá trị tài sản: KNN có thể được sử dụng để dự đoán giá trị của một tài sản dựa trên các thông tin về các tài sản tương tự.
– Phát hiện gian lận: KNN có thể được sử dụng để phát hiện gian lận trong các giao dịch tài chính hoặc các hình thức giao dịch khác.
– Phân tích dữ liệu khách hàng: KNN có thể được sử dụng để phân tích dữ liệu khách hàng và dự đoán hành vi của khách hàng.

- Tìm kiếm sản phẩm tương tự: KNN có thể được sử dụng để tìm kiếm các sản phẩm tương tự trên các trang thương mại điện tử.
- Phân loại văn bản: KNN có thể được sử dụng để phân loại văn bản, chẳng hạn như phân loại email vào các thư mục khác nhau.

Đề xuất sản phẩm: KNN có thể được sử dụng để đề xuất các sản phẩm tương tự cho người dùng, chẳng hạn như các sản phẩm liên quan đến sản phẩm mà người dùng đã mua trước đó.

CHƯƠNG 3: ỨNG DỤNG CÁC KỸ THUẬT PHÂN LỚP DỮ LIỆU NHẰM PHÁT HIỆN HÌNH THỨC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN(DDoS)

3.1. Giới thiệu

Mạng Hướng Phần Mềm (Software-Defined Networking - SDN) là một mô hình mới giúp quản lý mạng hiệu quả hơn với kiến trúc linh hoạt và có thể lập trình được. Trong SDN, các lĩnh vực điều khiển và quản lý dữ liệu được phân chia độc lập và được quản lý bởi một bộ điều khiển trung tâm. Do đó, người quản trị có thể dễ dàng áp dụng chính sách mạng khác nhau trên toàn mạng chỉ từ một vị trí duy nhất. Hình 1 mô tả cấu trúc phân lớp của môi trường SDN. Tuy nhiên, mô hình mới này cũng đem lại các vấn đề về bảo mật. SDN bị phơi nhiễm với các cuộc tấn công độc hại dành riêng cho nó, bên cạnh các cuộc tấn công truyền thống. Cuộc tấn công nghiêm trọng nhất đối với SDN là cuộc tấn công vào bộ điều khiển, vì kẻ tấn công có khả năng kiểm soát và điều khiển toàn bộ mạng hoặc tạo sự cố. Các cuộc tấn công DDoS là mối đe dọa chính đối với bộ điều khiển vì chúng có thể làm người dùng không thể truy cập được các dịch vụ mạng.

Các cuộc tấn công DDoS được thực hiện bằng số lượng lớn máy tính từ nhiều vị trí khác nhau, gây khó khăn cho việc phát hiện và ngăn chặn. Tần suất và mức độ nghiêm trọng của các cuộc tấn công DDoS không ngừng tăng lên, gây ảnh hưởng nghiêm trọng đến nhiều dịch vụ mạng. Vì vậy, việc phát hiện và ngăn chặn các cuộc tấn công DDoS nhanh chóng là một trong những vấn đề quan trọng nhất đối với các nhà cung cấp dịch vụ mạng và quản trị viên. Các lớp SDN khác nhau có thể bị vô hiệu hóa bằng cách lấp đầy các kênh giao tiếp giữa bộ điều khiển và bộ chuyển mạch hoặc giữa bộ điều khiển và lớp ứng dụng bằng thông tin luồng không cần thiết từ các cuộc tấn công DDoS. Thông thường, khi thiết kế, các nhà phát triển hệ thống không tích hợp cơ chế bảo mật trên bộ điều khiển có thể phân biệt được giữa lưu lượng tấn công và lưu lượng bình thường. Do đó, rất khó phát hiện ra các cuộc tấn công nhắm vào các SDN.

3.2. Xây dựng bộ dữ liệu

3.2.1. Mô tả bộ dữ liệu

Đây là một bộ dữ liệu cụ thể về SDN được tạo ra bằng cách sử dụng phần mềm giả lập Mininet và được sử dụng để phân loại lưu lượng mạng bằng các thuật toán học máy và học sâu. Quá trình xây dựng cơ sở dữ liệu bắt đầu bằng cách tạo mười cấu trúc liên kết trong mạng Mininet, trong đó các thiết bị chuyển mạch được kết nối với một bộ điều khiển duy nhất là Ryu. Mô phỏng mạng chạy với lưu lượng TCP, UDP và ICMP bình thường và lưu lượng độc hại, bao gồm các tấn công TCP Syn, tấn công UDP Flood và tấn công ICMP. Tổng cộng, tập dữ liệu này chứa 23 đặc trưng khác nhau, trong đó một số đặc trưng được trích xuất từ các thiết bị chuyển mạch và một số đặc trưng khác được tính toán từ các thông số khác. Các đặc trưng này bao gồm Switch-id, Packet_count, byte_count, duration_sec, duration_nsec (đo lường thời gian bằng nano giây), total_duration (tổng thời gian tính bằng giây và nano giây), địa chỉ IP nguồn, địa

chỉ IP đích, số cổng, tx_bytes (số byte được chuyển đi từ cổng chuyển mạch), rx_bytes (số byte được nhận trên cổng chuyển mạch), ngày giờ (đã được chuyển đổi thành số) và luồng (được theo dõi trong khoảng thời gian giám sát là 30 giây). Ngoài ra, các đặc trưng khác được tính toán bao gồm Packet per flow (số gói trong một luồng đơn lẻ), Byte per flow (số byte trong một luồng đơn lẻ), Packet rate (số gói được gửi mỗi giây), số tin nhắn Packet_ins, tổng số mục lưu lượng trong switch, tx_kbps, rx_kbps (tốc độ truyền và nhận dữ liệu) và port bandwitdh (tổng của tx_kbps và rx_kbps).

Phần cuối cùng của bảng thể hiện nhãn lớp cho biết loại lưu lượng truy cập là độc hại hay lành tính. Lưu lượng lành tính được đánh dấu là 0 và lưu lượng độc hại được đánh dấu là 1. Mô phỏng mạng đã chạy trong vòng 250 phút và đã thu thập được 104.345 mẫu dữ liệu. Mô phỏng này được thực hiện trong một khoảng thời gian xác định và có thể tiếp tục thu thập thêm dữ liệu.

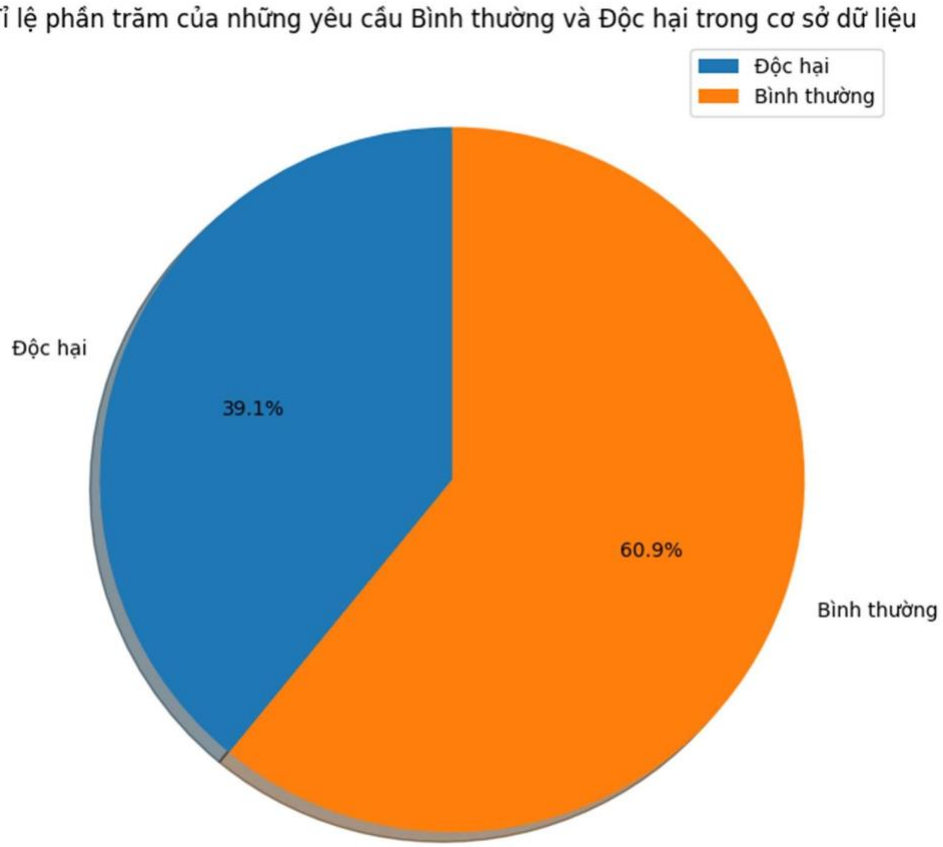
STT	Tên đặc trưng	Mô tả	Kiểu dữ liệu
1	dt	Ngày giờ	int64
2	switch	Mã số của bộ chuyển mạch	int64
3	src	Địa chỉ nguồn	object
4	dst	Địa chỉ đích	object
5	ptkcount	Số lượng gói tin	int64
6	bytecount	Số lượng byte	int64
7	dur	Thời lượng được tính theo giây	int64
8	dur_nsec	Thời lượng được tính theo nano giây	int64
9	tot_dur	Tổng thời lượng của dur và dur_nsec	float64
10	flows	Số luồng	int64
11	packetins	Số tin nhắn	int64
12	ptkperflow	Số gói trên mỗi luồng đơn	int64
13	byteperflow	Số byte trên mỗi luồng đơn	int64
14	ptkrate	Số gói tin truyền đi trong mỗi giây	int64
15	Pairflow	Số cặp luồng	int64
16	Protocol	Giao thức được thực hiện	object
17	port_no	Số cổng truyền tin	int64
18	tx_bytes	Số lượng gói tin truyền đi từ các cổng bộ chuyển mạch	int64
19	rx_bytes	Số lượng gói tin nhận về ở cổng bộ chuyển mạch	int64
20	tx_kbps	Thông lượng truyền đi qua cổng bộ chuyển mạch	int64
21	rx_kbps	Thông lượng nhận về qua cổng bộ chuyển mạch	float64
22	tot_kbps	Tổng thông lượng	float64

23	label	Nhãn	int64

Bảng 2: Mô tả bộ dữ liệu

Các bản ghi được gán nhãn hoặc là Bình thường (0) hoặc là Độc hại (1):

- Số bản ghi được gán nhãn bình thường là: 63561
- Số bản ghi được gán nhãn độc hại là: 40784



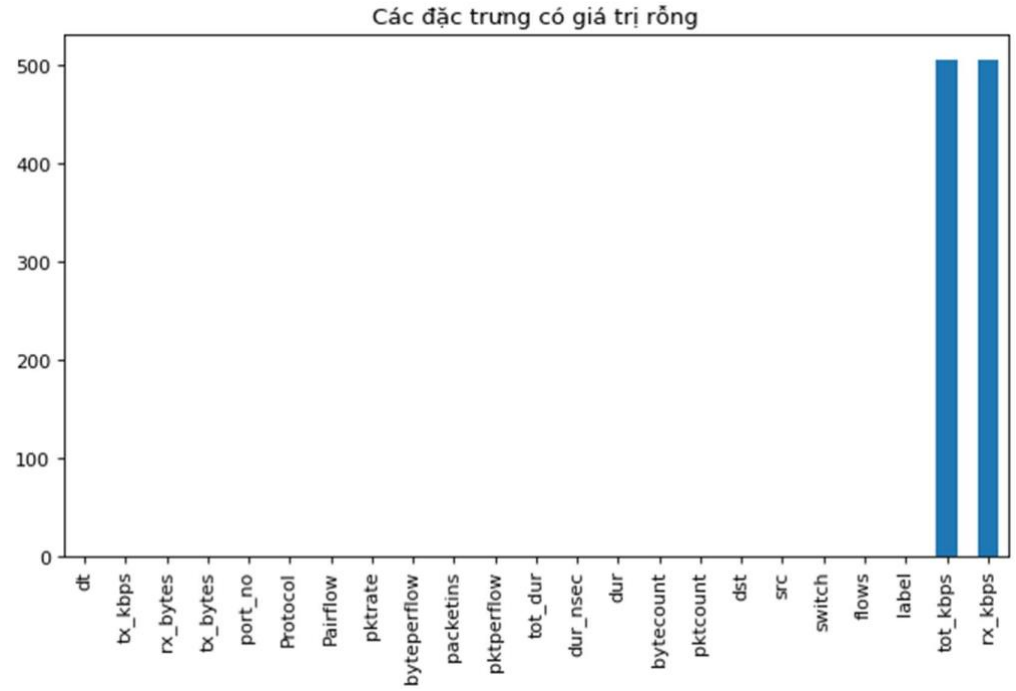
Hình 16: Tỉ lệ phần trăm của những yêu cầu Bình thường và Độc hại trong cơ sở dữ liệu

dt	switch	src	dst	pktcount	bytecount	dur	dur_nsec	tot_dur	flows	packetins	pktperflow	byteperflc	pktrate	Pairflow	Protocol	port_no	tx_bytes	rx_bytes	tx_kbps	rx_kbps	tot_kbps	label
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	3	1.4E+08	3917	0	0	0	0
11605	1	10.0.0.1	10.0.0.8	126395	1.3E+08	280	7.3E+08	2.81E+11	2	1943	13531	1.4E+07	451	0	UDP	4	3842	3520	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	1	3795	1242	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	2	3688	1492	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	3	3413	3665	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	1	3795	1402	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	4	3665	3413	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	1	3775	1492	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	3845	1402	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	4	3.5E+08	4295	16578	0	16578	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	1	3775	1242	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	2	3413	3665	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	1	4047	1.4E+08	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	4	3665	3413	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	5.8E+08	2586	19164	0	19164	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	4	3665	3413	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	3795	1402	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	3795	1492	0	0	0	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	4047	1.9E+08	0	6307	6307	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	1	3879	4.8E+07	0	3838	3838	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	2	4055	9.6E+07	0	3838	3838	0
11425	1	10.0.0.1	10.0.0.8	45304	4.8E+07	100	7.2E+08	1.01E+11	3	1943	13535	1.4E+07	451	0	UDP	3	3413	3665	0	0	0	0
11425	1	10.0.0.2	10.0.0.8	90333	9.6E+07	200	7.4E+08	2.01E+11	3	1943	13534	1.4E+07	451	0	UDP	1	3570	1492	0	0	0	0
11455	1	10.0.0.2	10.0.0.8	103866	1.1E+08	230	7.5E+08	2.31E+11	3	1943	13533	1.4E+07	451	0	UDP	1	3775	1242	0	0	0	0
11605	1	10.0.0.1	10.0.0.8	126395	1.3E+08	280	7.3E+08	2.81E+11	2	1943	13531	1.4E+07	451	0	UDP	3	4766	3.5E+08	0	6400	6400	0
11515	1	10.0.0.1	10.0.0.8	85676	9.1E+07	190	7.3E+08	1.91E+11	3	1943	13306	1.4E+07	443	0	UDP	2	3795	1492	0	0	0	0
11515	1	10.0.0.1	10.0.0.8	85676	9.1E+07	190	7.3E+08	1.91E+11	3	1943	13306	1.4E+07	443	0	UDP	4	5E+08	4715	12831	0	12831	0
11515	1	10.0.0.1	10.0.0.8	85676	9.1E+07	190	7.3E+08	1.91E+11	3	1943	13306	1.4E+07	443	0	UDP	3	2.3E+08	4169	7676	0	7676	0

Hình 17: Các bản ghi trong cơ sở dữ liệu

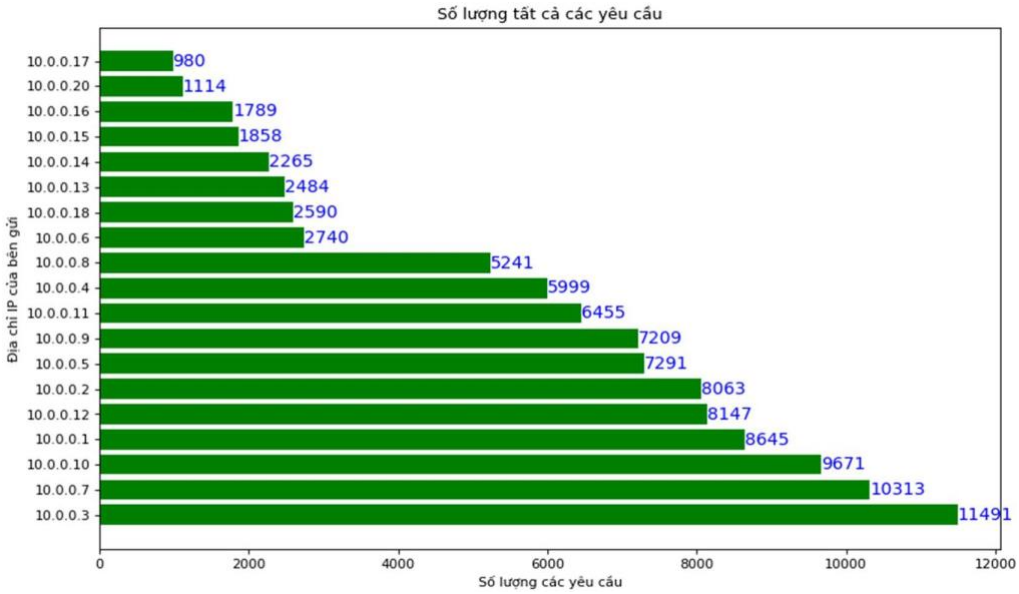
3.2.2. Phân tích đặc trưng trong cơ sở dữ liệu

Kiểm tra số các đặc trưng chứa các giá trị NULL nhằm loại bỏ các giá trị NULL, tạo điều kiện thuận lợi cho việc xây dựng các mô hình phân lớp. Kết quả kiểm tra giá trị NULL cho thấy, có 2 đặc trưng có chứa các giá trị NULL là tot_kbps và rx_kbps.



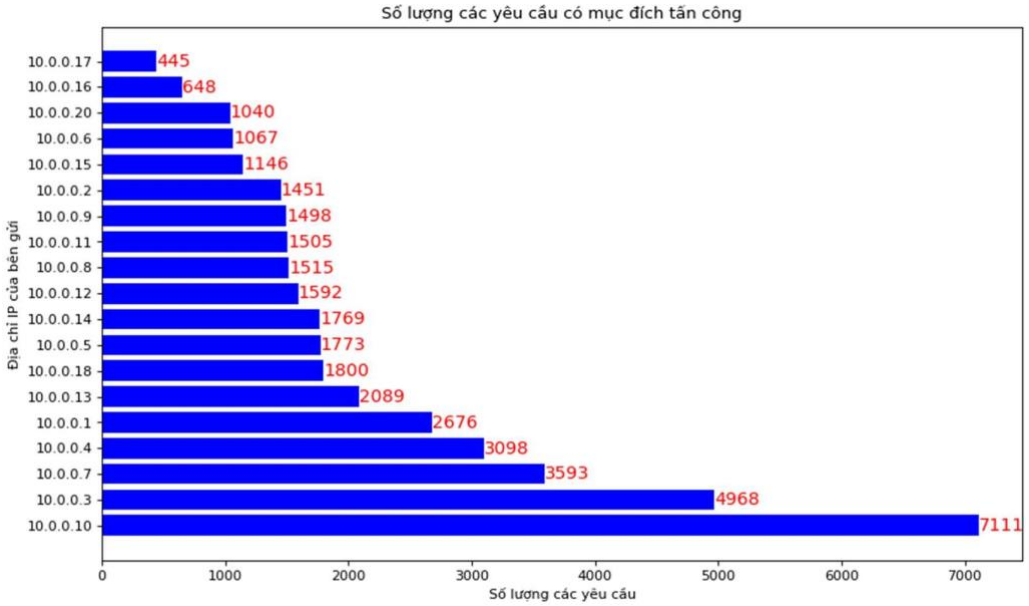
Hình 18: Biểu đồ các đặc trưng rỗng

Tiếp theo, đánh giá về thống kê của các địa chỉ gửi dữ liệu và số lượng các yêu cầu (request) tương ứng với các địa chỉ máy gửi trong cơ sở dữ liệu.



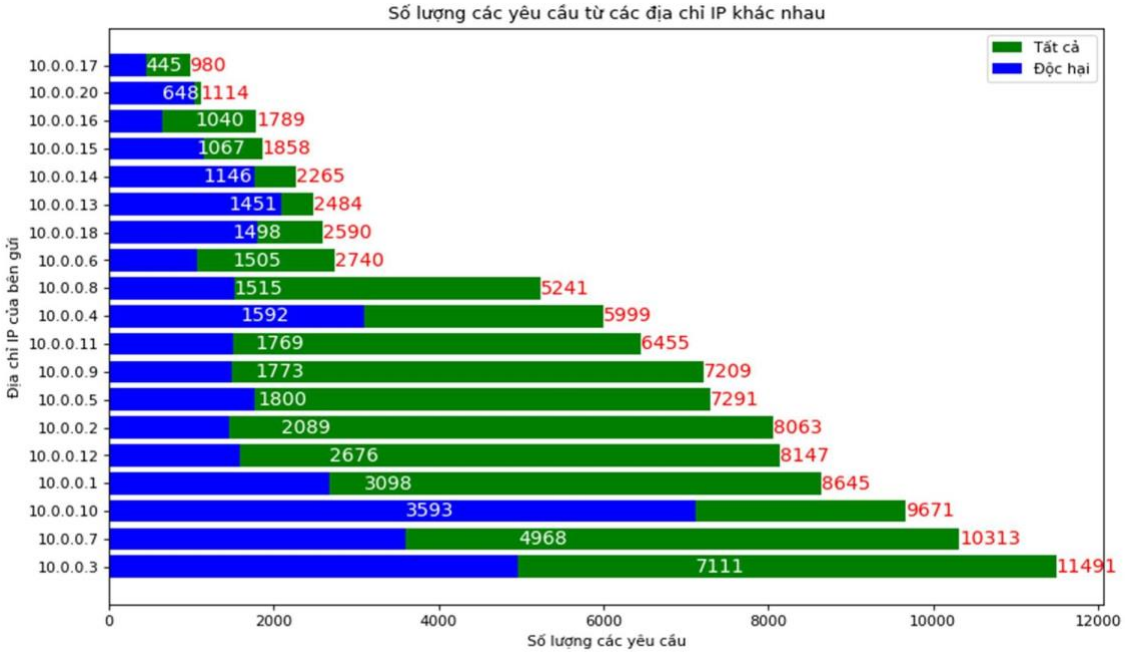
Hình 19: Biểu đồ phân phối của các địa chỉ gửi và số lượng yêu cầu được gửi đi

Căn cứ vào cơ sở dữ liệu, nhãn được gán cho các bản ghi, tiếp tục đánh giá và kiểm tra trong số những yêu cầu được gửi đi có bao nhiêu những yêu cầu là độc hại, có mục đích tấn công vào hệ thống.



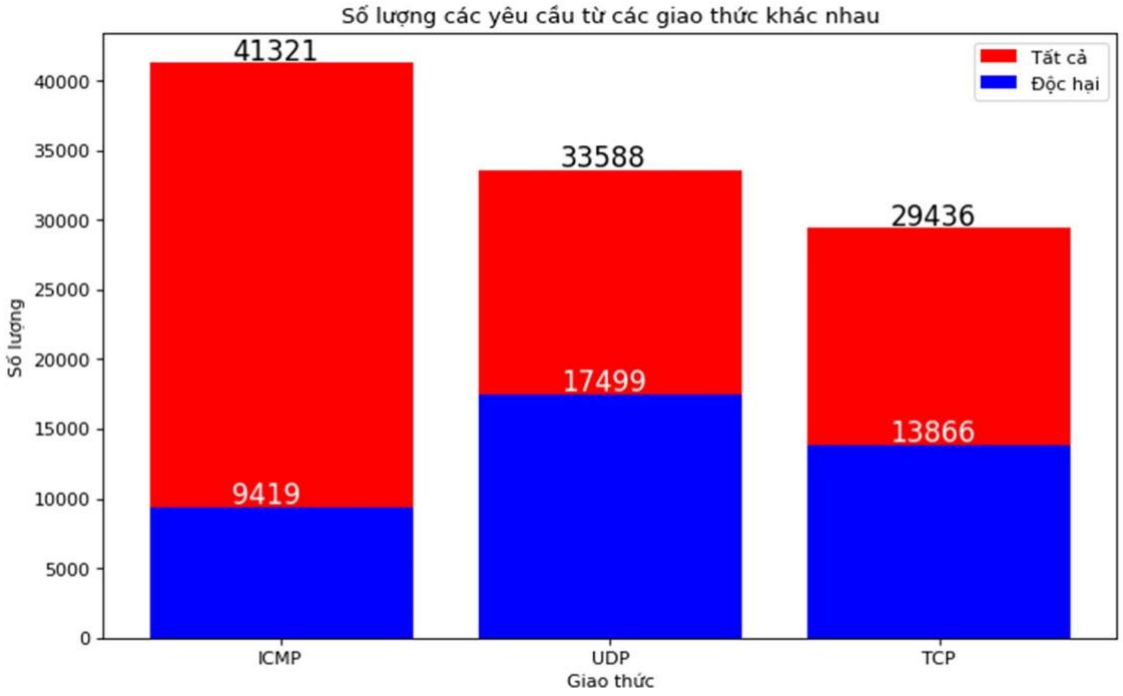
Hình 20: Biểu đồ về số lượng các yêu cầu có mục đích tấn công đối với các địa chỉ gửi

Đã đánh giá tỉ lệ các số lượng yêu cầu có mục đích tấn công so với tổng số các yêu cầu gửi đến các hệ thống. Kết quả được thể hiện trong Hình *.



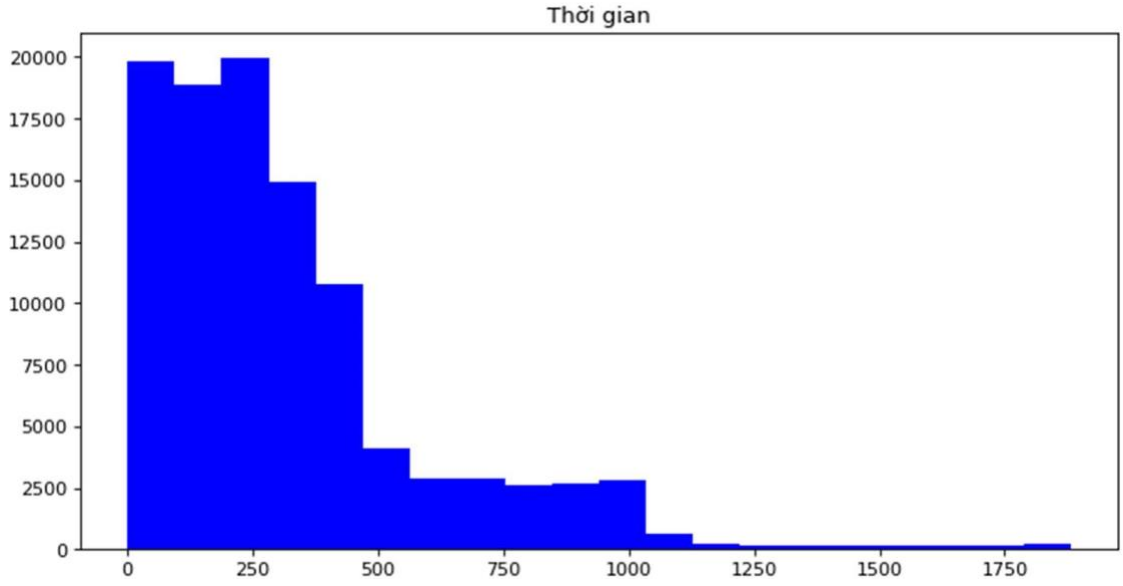
Hình 21: Biểu đồ về số lường các yêu cầu gửi từ các địa chỉ khác nhau

Đối với các giao thức khác nhau, khả năng bảo mật trước các tấn công DDOS cũng khác nhau, trong quá trình xây dựng cơ sở dữ liệu, các tác giả đã sử dụng 3 phương thức là ICMP, UDP, TCP. Để đánh giá mức độ ảnh hưởng của các giao thức với mức độ tấn công, tác giả thực hiện việc thống kê số lượng các yêu cầu từ các giao thức khác nhau và được đã được phân lớp.



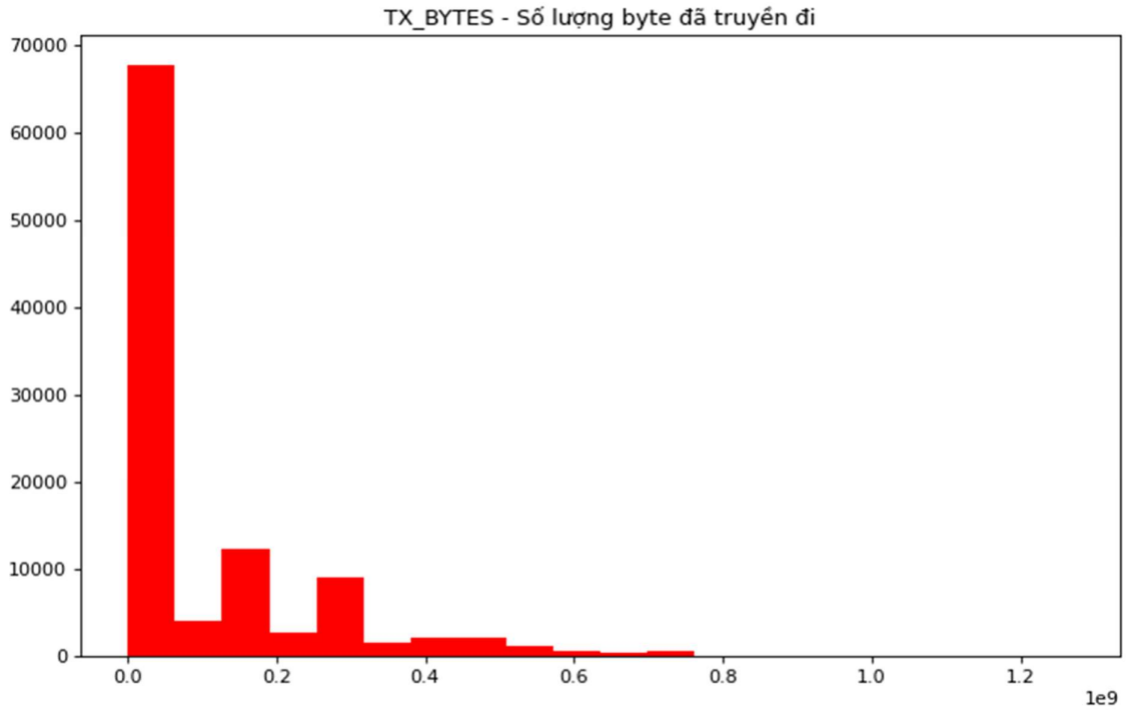
Hình 22 : Biểu đồ phân bố về các yêu cầu và các giao thức được sử dụng

Thực hiện việc thống kê về thời lượng của các luồng hoạt động trên hệ thống.

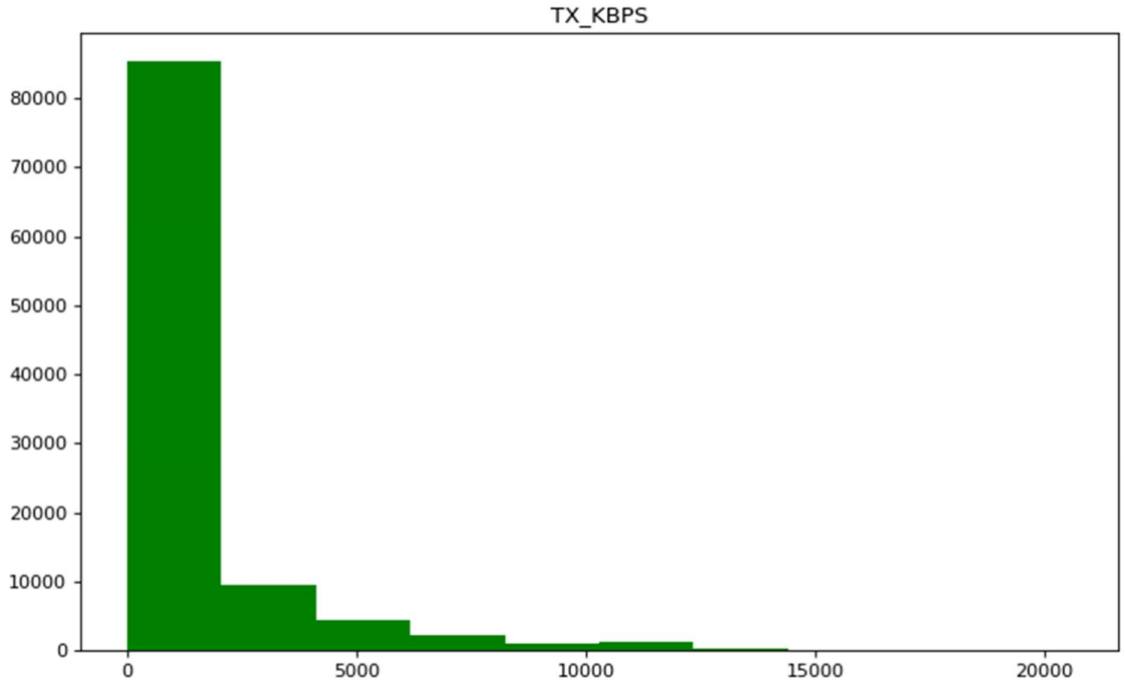


Hình 23: Biểu đồ phân bố về thời gian của các yêu cầu

Số lượng dữ liệu truyền đi theo byte cũng được thống kê nhằm đánh giá về sự khác nhau giữa các luồng an toàn và luồng tấn công.

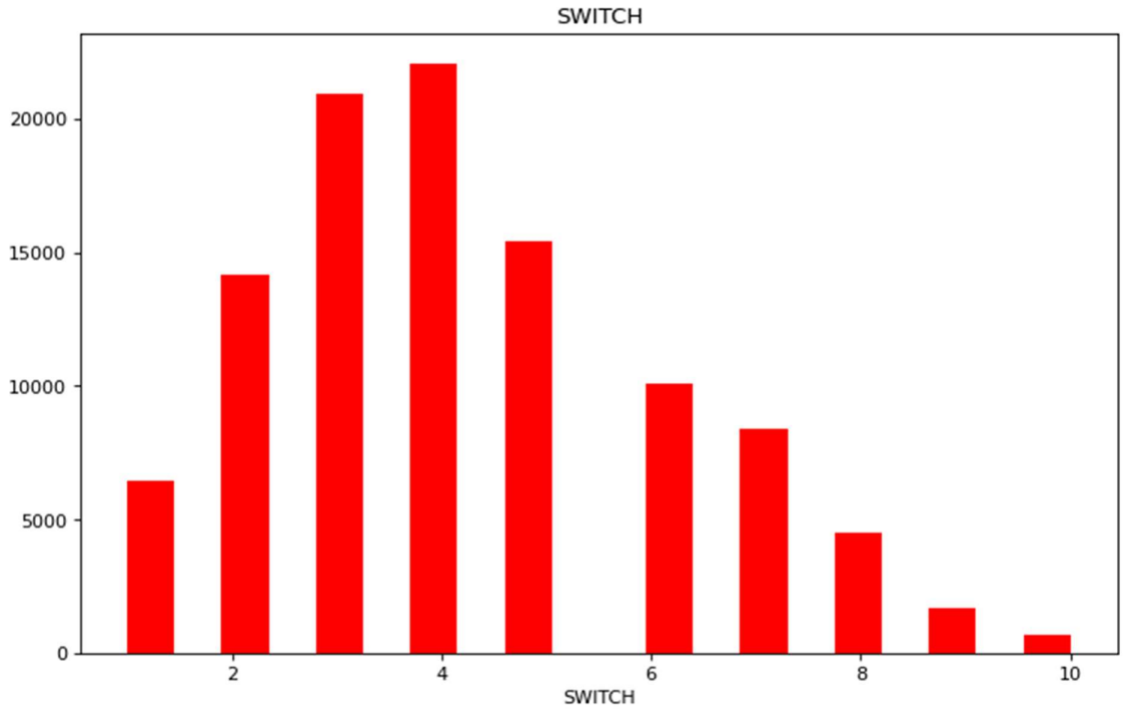


Hình 24: Biểu đồ phân bố số lượng các yêu cầu và kích thước dữ liệu truyền đi

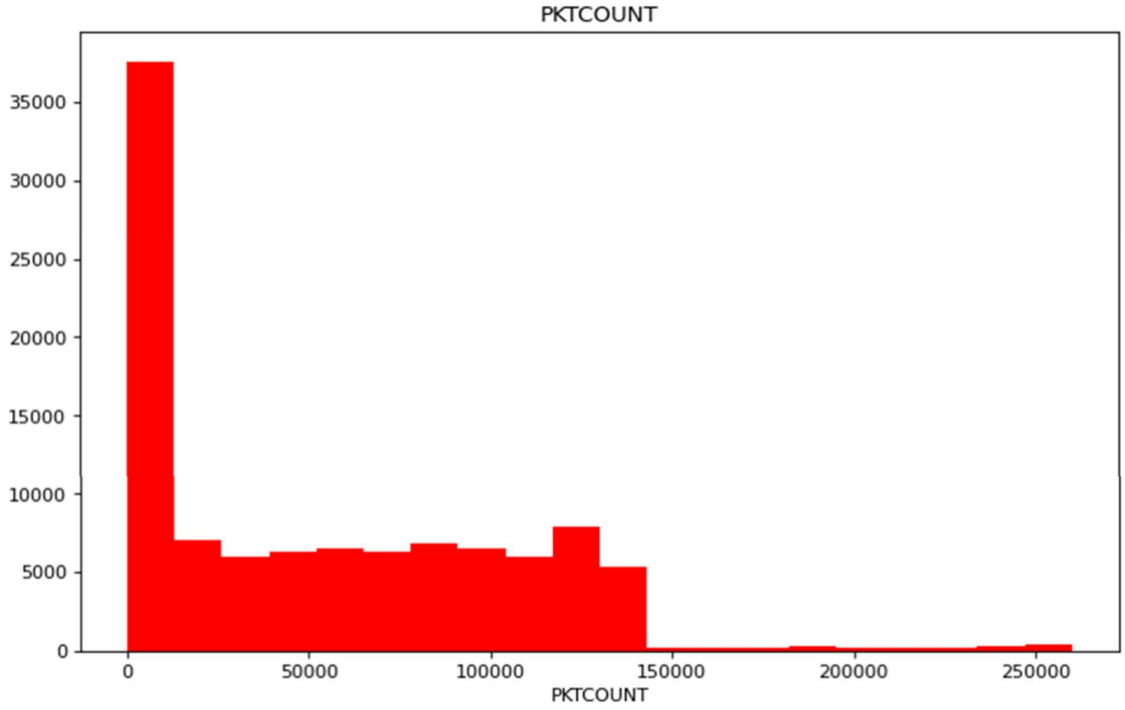


Hình 25: Biểu đồ phân bố số lượng các yêu cầu và băng thông dữ liệu

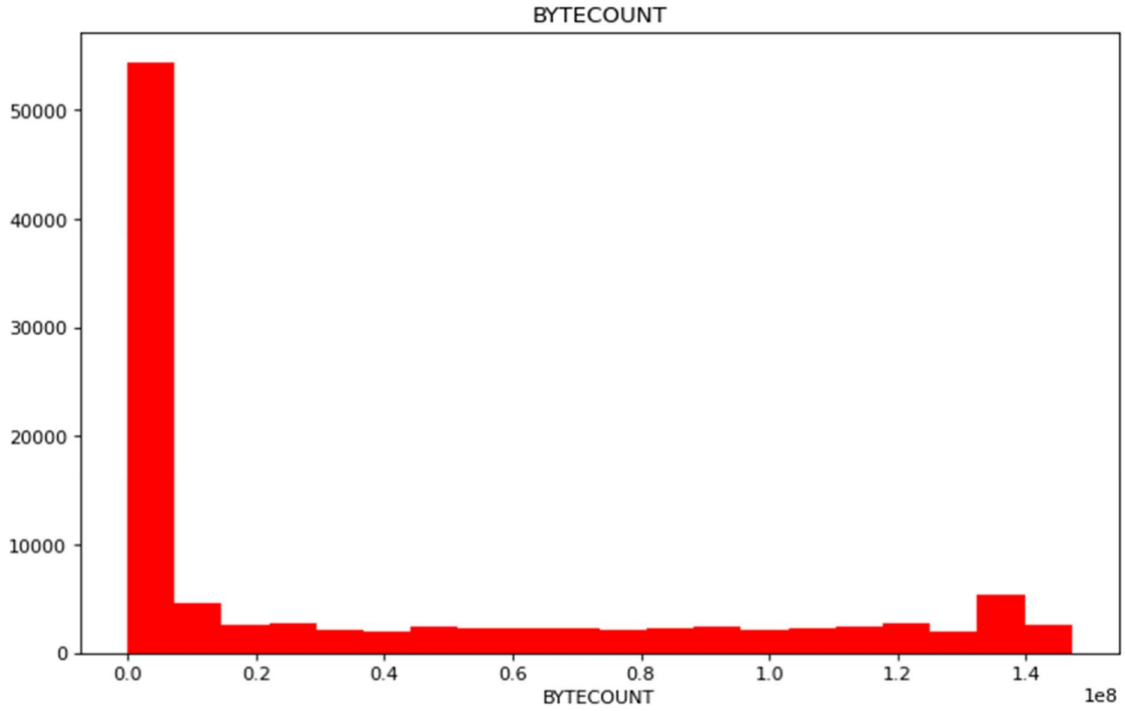
Số lượng các bộ chuyển mạch, thành phần bộ chuyển mạch cũng được đánh giá để xem mức độ ảnh hưởng của các bộ chuyển mạch đối với lượng yêu cầu được gửi đến.



Hình 26: Biểu đồ phân bố số lượng các yêu cầu và bộ chuyển mạch



Hình 27: Biểu đồ phân bố số lượng các yêu cầu và số lượng gói tin



Hình 28: Biểu đồ phân bố số lượng các yêu cầu và số byte dữ liệu

Việc đánh giá các đặc trưng dựa trên dữ liệu hiện có cho phép chúng ta có được cái nhìn tổng quan về sự ảnh hưởng của các đặc trưng và số lượng các yêu cầu được gửi đến. Các yêu cầu có thể là bình thường, cũng có thể là các yêu cầu mang mục đích tấn công.

3.2.3. Trích chọn đặc trưng

Mục đích chính của việc lựa chọn đặc trưng là việc lựa chọn ra một tập các đặc trưng nhằm giảm được chi phí tính toán và giảm được mối quan hệ giữa các đặc trưng không liên quan đến tập các đặc trưng có thể ảnh hưởng đến hiệu năng hoạt động của mô hình. Chúng tôi sử dụng giải thuật NCA(Neighbourhood Component Analysis) để tìm kiếm các đặc trưng phù hợp nhất cho việc xây dựng mô hình phân lớp của hơn 100 nghìn bản ghi, trong đó có 22 đặc trưng liên quan đến mạng SDN. Ưu điểm của giải thuật NCA là được phát triển trên giải thuật kNN, cho phép liệt kê ra các đặc trưng theo mức độ quan trọng và cũng cung cấp được thông tin về trọng số của các đặc trưng.

Có khả năng một tính năng được đưa ra làm đầu vào $\boldsymbol{x}_i$ trong thuật toán NCA tương ứng với lớp y_i, tương ứng với tất cả các lớp .Khoảng cách giữa hai lần quan sát được tính theo công thức sau:

$$d_w = \Sigma_r^p w^2 |x_{ir} - x_{jr}|$$

Trong đó, w_r là trọng số của đặc trưng .Điểm tham chiếu (P) trong tập các đặc trưng được tính theo công thức sau:

$$P(\text{Ref}(x_i) = x_j \mid S) = \frac{k(d_w(x_i, x_j))}{\sum_{j=1}^{N} k(d_w(x_i, x_j))}$$

Xác suất để chọn x_i là một điểm tham chiếu cho x_j được tính như sau:

$$P_{ij}(\text{Ref}(x_i) = x_j \mid S) = \frac{k(d_w(x_i, x_j))}{\sum_{j=1Gi}^{N} k(d_w(x_i, x_j))}$$

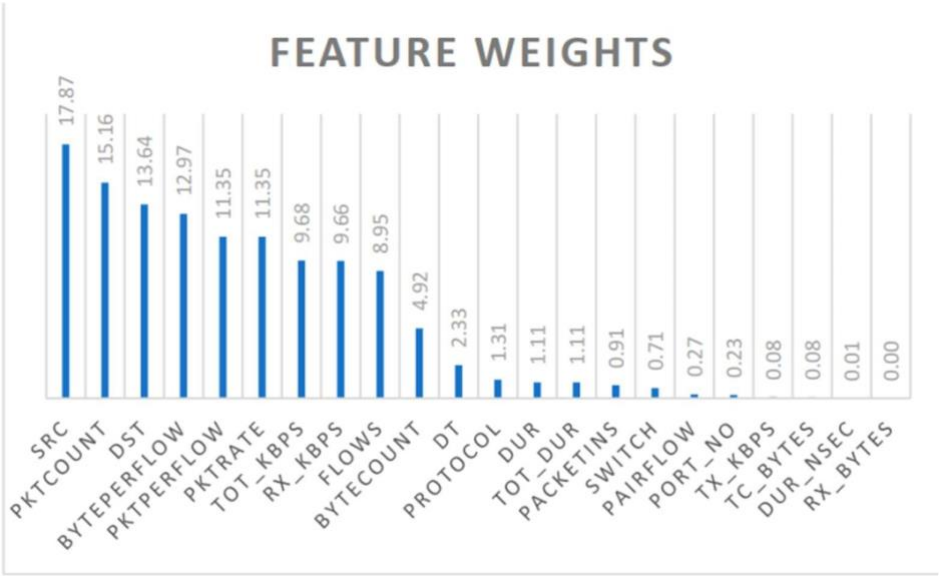
Ở đây, k tương đương với hàm lõi (k(z) = (exp – z/σ) và σ biểu diễn cho chiều rộng của hàm lõi trong đó khả năng phân lớp đúng của lớp thực được tính theo công thức sau:

$$P_i = \sum_j y_i P_{ij}$$

Trong đó P_{ij} = 1 khi và chỉ khi $y_i = y_j$ vày_{ij} = 0

Dữ liệu mạng thô trong tập dữ liệu được thực hiện ở bước tiền xử lý và lựa chọn tính năng được áp dụng với thuật toán NCA. Để huấn luyện thuật toán NCA, giá trị tham số chính quy lambda (λ), ngăn chặn việc trang bị quá mức, đã được xác định tự động. Phương pháp giảm độ dốc ngẫu nhiên (SGD) được sử dụng để tối ưu hóa trọng số của đối tượng địa lý. Trong tối ưu hóa SGD, giá trị kích thước minibatch được xác định là 10 và giá trị epoach là 5. Trong khi giá trị trọng số của các đối tượng không liên quan trong thuật toán NCA gần bằng 0, giá trị trọng số của các đối tượng có đặc điểm phân biệt cao sẽ cao hơn.

Khi xem xét giá trị trọng số của các đặc trưng bằng thuật toán NCA, chúng tôi nhận thấy rằng giá trị trọng số của 8 đặc trưng nằm trong khoảng từ 0 đến 1, trong khi giá trị trọng số của 14 đặc trưng thay đổi từ 1.11 đến 17.87. Các thuật toán học máy được biết là có ảnh hưởng đến chi phí tính toán khi phân loại các vấn đề đặc tả cao. Vì lý do này, sau khi phân tích 22 tính năng mạng các thuật toán NCA, quá trình phân loại đầu tiên được thực hiện với 8 tính năng có giá trị chỉ số lớn hơn 9. Trong nghiên cứu thử nghiệm thứ hai, 14 tính năng hiệu quả đã được chọn và đưa ra làm dữ liệu đầu vào cho các thuật toán học máy. Danh sách 14 thuộc tính và giá trị trọng lượng hiệu quả nhất được NCA lựa chọn được thể hiện trong Bảng 2



Hình 29: Trọng số các đặc trưng được tính theo giải thuật NCA

Đặc trưng	Trọng số được tính theo NCA
src	17,87
pktcount	15,16
dst	13,64
byteperflow	12,97
pktperflow	11,35
pktrate	11,35
tot_kbps	9,68
rx_kbps	9,66
flows	8,95
bytecount	4,92
dt	2,33
protocol	1,31
dur	1,11
tot_dur	1,11

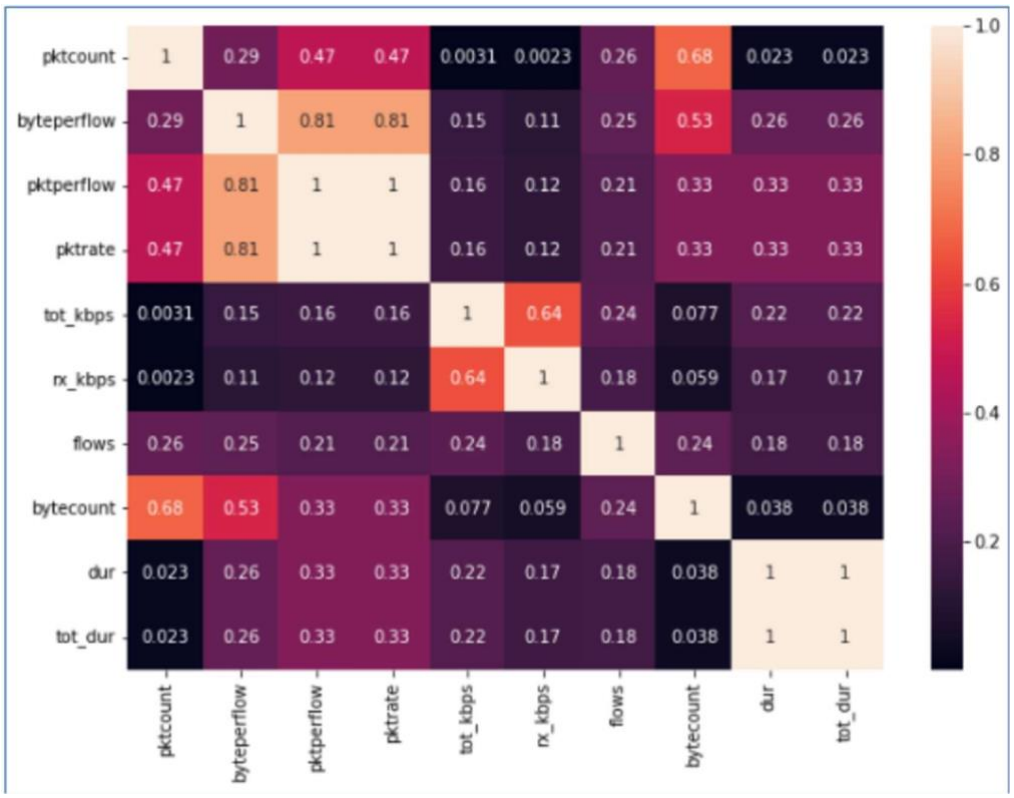
Bảng 3: Trọng số được tính theo NCA

Bởi vì các đặc trưng như src, dst, dt là không cần thiết cho việc triển khai mô hình nên ta loại bỏ các đặc trưng này khỏi danh sách các đặc trưng lực chọn. Khi đó các đặc trưng còn lại là:

	pktcount	byteperflow	pktperflow	pktrate	tot_kbps	rx_kbps	flows	bytecount	Protocol	dur	tot_dur
0	45304	14428310	13535	451	0.0	0.0	3	48294064	UDP	100	1.010000e+11
1	126395	14424046	13531	451	0.0	0.0	2	134737070	UDP	280	2.810000e+11
2	90333	14427244	13534	451	0.0	0.0	3	96294978	UDP	200	2.010000e+11
3	90333	14427244	13534	451	0.0	0.0	3	96294978	UDP	200	2.010000e+11
4	90333	14427244	13534	451	0.0	0.0	3	96294978	UDP	200	2.010000e+11
5	90333	14427244	13534	451	0.0	0.0	3	96294978	UDP	200	2.010000e+11
6	45304	14428310	13535	451	0.0	0.0	3	48294064	UDP	100	1.010000e+11
7	45304	14428310	13535	451	0.0	0.0	3	48294064	UDP	100	1.010000e+11
8	45304	14428310	13535	451	0.0	0.0	3	48294064	UDP	100	1.010000e+11
9	90333	14427244	13534	451	16578.0	0.0	3	96294978	UDP	200	2.010000e+11

Bảng 4:Trọng số các đặc trưng còn lại sau khi loại bỏ

Sau đó, chúng tôi tính mức độ tương quan giữa các đặc trưng bằng hàm tính tương quan đã được xây dựng sẵn của Python nhằm xác định mức độ phụ thuộc của các đặc trưng.



Hình 30: Mức độ tương quan giữa các đặc trưng đã được lựa chọn trọng cơ sở dữ liệu

Khi xem xét bảng tương quan, ta thấy có một số đặc trưng là trùng nhau hoặc có độ tương quan lớn như “dur” và “tot_dur”, “pktperflow” và “pktrate”, do vậy, ta loại bỏ 3 đặc trưng là 'dur', "pktrate", "pktperflow ra khỏi bảng dữ liệu. Khi đó ta có:

Bằng cách này, chúng tôi lựa chọn được các đặc trưng phục vụ quá trình xây dựng mô hình dự đoán tấn công DDOS vào các mạng SDN dựa trên cơ sở dữ liệu đã được xây dựng.

3.3. Triển khai các mô hình không sử dụng giải thuật lựa chọn đặc trưng

Khi không sử dụng giải thuật lựa chọn đặc trưng, tác giả thực hiện việc ứng dụng các mô hình phân lớp vào cơ sở dữ liệu với 19 đặc trưng (loại bỏ 4 đặc trưng không có ý nghĩa và nhãn là "dt", "src", "dst", "label". Kết quả triển khai cho thấy, độ chính xác cao nhất khi sử dụng mô hình Random Forest, với độ chính xác là 99.99%.

- Sử dụng mô hình phân lớp cây quyết định (Decision Tree):
  The Accuracy is : 98.22%

	precision	recall	f1-score	support
0	0.98	0.99	0.99	18743
1	0.99	0.97	0.98	12409
accuracy			0.98	31152
macro avg	0.98	0.98	0.98	31152
weighted avg	0.98	0.98	0.98	31152

- Sử dụng kỹ thuật Logistic Regression
  Accuracy: 76.64%
  Best solver is : liblinear

	precision	recall	f1-score	support
0	0.84	0.79	0.81	20024
1	0.66	0.72	0.69	11128
accuracy			0.77	31152
macro avg	0.75	0.76	0.75	31152
weighted avg	0.77	0.77	0.77	31152

- Sử dụng kỹ thuật Random Forest
  Accuracy of RF is : 99.99%

	precision	recall	f1-score	support
0	1.00	1.00	1.00	18984
1	1.00	1.00	1.00	12168
accuracy			1.00	31152
macro avg	1.00	1.00	1.00	31152
weighted avg	1.00	1.00	1.00	31152

- Sử dụng Support Vector Machine
  Accuracy of SVM model 97.0%
  Best kernel is : rbf

	precision	recall	f1-score	support
0	0.97	0.98	0.97	18750
1	0.97	0.95	0.96	12402
accuracy			0.97	31152
macro avg	0.97	0.96	0.97	31152
weighted avg	0.97	0.97	0.97	31152

3.3. Triển khai các mô hình có sử dụng giải thuật lựa chọn đặc trưng

Ứng dụng kỹ thuật lựa chọn đặc trưng vào cơ sở dữ liệu, khi đó, tác giả thực hiện việc xây dựng mô hình với 7 đặc trưng có ý nghĩa quan trọng với các thông tin về mạng SDN. Đó là các đặc trưng: pktcount, byteperflow, tot_kbps, rx_kbps, flows, bytecount, tot_dur. Kết quả triển khai cho thấy, độ chính xác cao nhất khi sử dụng mô hình Random Forest, với độ chính xác là 99.42%.

- Sử dụng kỹ thuật Logistic Regression
  Accuracy: 75.20%
  Best solver is : sag

	precision	recall	f1-score	support
0	0.85	0.77	0.81	20965
1	0.60	0.72	0.65	10187
accuracy			0.75	31152
macro avg	0.72	0.74	0.73	31152
weighted avg	0.77	0.75	0.76	31152

– Sử dụng kỹ thuật Random Forest
Accuracy of RF is : 99.42%

	precision	recall	f1-score	support
0	0.99	1.00	1.00	18922
1	1.00	0.99	0.99	12230
accuracy			0.99	31152
macro avg	0.99	0.99	0.99	31152
weighted avg	0.99	0.99	0.99	31152

– Sử dụng kỹ thuật Decision Tree

	precision	recall	f1-score	support
0	0.91	1.00	0.95	17287
1	1.00	0.87	0.93	13865
accuracy			0.94	31152
macro avg	0.95	0.94	0.94	31152
weighted avg	0.95	0.94	0.94	31152

– Sử dụng kỹ thuật Support Vector Machine
Accuracy of SVM model 92.0%
Best kernel is : rbf

	precision	recall	f1-score	support
0	0.90	0.96	0.93	17287
1	0.95	0.86	0.90	13865
accuracy			0.92	31152
macro avg	0.92	0.91	0.91	31152
weighted avg	0.92	0.92	0.92	31152

KẾT LUẬN

1. Kết quả đạt được

Về mặt khoa học, tác giả đã thực hiện được những vấn đề sau:

- Nghiên cứu tổng quan cơ sở lý thuyết về an ninh mạng và hình thức tấn công DDOS hiện đang được sử dụng để tấn công các mạng dịch vụ.
- Phân tích, đánh giá được dữ liệu về các cuộc tấn công DDOS đã có dựa trên cơ sở dữ liệu có sẵn.
- Nghiên cứu tổng quan về các kỹ thuật trí tuệ nhân tạo được sử dụng để xây dựng các phân lớp, các mô hình dự đoán.

Về mặt thực tiễn, tác giả đã thực hiện được những việc sau

- Lựa chọn và triển khai được mô hình dự đoán, phát hiện được các tấn công DDOS dựa trên cơ sở dữ liệu hiện có, xác định được mô hình có độ chính xác cao nhất.

2. Hạn chế

Tuy nhiên, hạn chế của các kết quả là: chưa đánh giá đầy đủ về vai trò của các đặc trưng và ảnh hưởng của các đặc trưng đến độ chính xác của mô hình dự đoán đã lựa chọn.

3. Hướng phát triển

Phân tích, đánh giá và kiểm thử mô hình dựa trên việc tiếp tục đánh giá ảnh hưởng của các đặc trưng đối với độ chính sác của các mô hình dự đoán

Thanh Hóa, ngày 10 tháng 09 năm 2022

Hiệu trưởng	Đơn vị chủ trì	GV hướng dẫn	Trưởng nhóm
		Nguyễn Thế Cường	**Lê Xuân Quang**